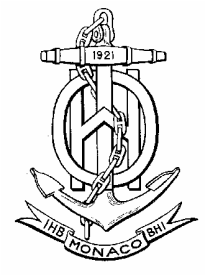
INTERNATIONAL HYDROGRAPHIC ORGANISATION

CHRIS19-06.3B



IHO DATA PROTECTION SCHEME

Edition 1.1 – August 2007

DRAFT

Special Publication No. 63

Published by the
International Hydrographic Bureau
MONACO

S-63

Page intentionally left blank

INTERNATIONAL HYDROGRAPHIC ORGANISATION

IHO DATA PROTECTION SCHEME

Edition 1.1 – August 2007

DRAFT

Special Publication S-63

Published by the
International Hydrographic Bureau
4, Quai Antoine 1er
B.P 445 - MC 98011 MONACO Cedex
Principality of Monaco
Tel: +(377) 93 10 81 00
Telefax: +(377) 93 10 81 40
E-mail: info@ihb.mc
Web: www.iho.shom.fr

Page intentionally left blank

PREFACE

Copyright infringement and data piracy are pervasive problems of the digital era. Electronic Navigational Charts (ENC) are not exempt from these issues. As well as the economic impact, the unofficial distribution of nautical information also gives rise to significant safety concerns. As a result, the publishers of official nautical information have sought to protect their data and provide the mariner with a certificate of authenticity through the adoption of a security schema.

In September 2000, IHO Member States were polled on their views on developing a single IHO Recommended Security Scheme (RSS) (see: IHB Circular Letter 38/2000). Responses indicated that a large majority of the Member States wished to have their ENC data encrypted and agreed that the IHO should adopt a single RSS (see: IHB CL 15/2001 Rev.1). A majority of the Member States responding also supported the adoption of the Primar Security Scheme as the IHO RSS, as it was at the time the de facto standard for ENC protection and the majority of ECDIS manufacturers had already developed the necessary decryption facilities in their systems.

The IHO Committee on Hydrographic Requirements for Information Systems (CHRIS), at its 13th meeting (Athens, Greece, September 2001), revisited the issue of a RSS and agreed that a small advisory expert group investigate the implications of IHB becoming the security scheme administrator for a RSS and assuming responsibility for the maintenance of a RSS.

The Data Protection Scheme Advisory Group (DPSWG) reported back to the IHB in January 2002 that there were no technical implications to the IHB becoming the security scheme administrator and that the level of effort to administer the security scheme would be limited and within the IHB resources. The DPSWG further provided a plan to develop an IHO RSS Version 1, based on the Primar Security Scheme. This Report was endorsed by CHRIS Members in February 2002 and the DPSWG was tasked to develop Version 1 of an IHO RSS.

The results were presented to CHRIS, at its 14th meeting (Shanghai, China, August 2002), which recommended that the ENC Security Scheme, as developed by the DPSWG, be submitted to IHO Member States for adoption as an IHO RSS, and that the role as Security Scheme Administrator be transferred to the IHB. These proposals (see: IHB CL 44/2002) were approved by a majority of Member States (see: IHB CL 66/2002). As a result, Edition 1.0 of the IHO Data Protection Scheme was issued in October 2003 as Publication S-63.

The 18th CHRIS meeting (Cairns, Australia, September 2006) tasked the DPSWG to develop a revised edition of S-63 with the following guidance:

- There would be no introduction of new features; changes would be kept to a minimum.
- Published S-63 guidelines would be included in the standard.
- S-63 would be reorganized to group issues specific to the IHB as Scheme Administrator, Data Servers, and OEMs.
- There would be a more precise description of the correct implementation of the IHO standard.

Accordingly, a draft Edition 1.1 of S-63 was prepared by DPSWG and endorsed by CHRIS at its 19th meeting (Rotterdam, Netherlands, November 2009). This was subsequently endorsed by Member States. Edition 1.1 therefore supersedes the previous edition.

This new edition includes supporting documentation, test data and a method to supply ENCs using “Large Media Support”. Changes to this Standard, as well as any further developments, will be coordinated by the DPSWG under CHRIS Guidance.

The IHB Directing Committee thanks all the contributors to this new edition of the IHO Data Protection Scheme; especially from DPSWG members, the Electronic Chart Centre of Norway, and the United Kingdom Hydrographic Office.

Captain Robert WARD
Director, IHB

Page intentionally left blank

TABLE OF CONTENTS

Page intentionally left blank

GLOSSARY

Glossary of S-63 Data Protection Scheme Terms

Blowfish	Encryption algorithm used by the protection scheme
Cell Key	Key used to produce encrypted ENC, and required to decrypt the encrypted ENC information.
Cell Permit	Encrypted form of Cell key, created specifically for a particular user.
Data Client	Term used to represent an end-user receiving the encrypted ENC information. The Data Client will be using a software application (e.g. ECDIS) to perform many of the operations detailed within the scheme. Typically, an ECDIS user.
Data Server	Term used to represent an organisation producing encrypted ENCs or issuing Cell Permits to end-users.
M_ID	The unique identifier assigned by the SA to each manufacture. Data Servers use this to identify which M_KEY to use when decrypting the Userpermit.
M_KEY	ECDIS manufacturer's unique identification key provided by the Scheme Administrator to the OEM. It is used by OEMs to encrypt the HW_ID when creating a userpermit.
HW_ID	The unique identifier assigned by an OEM to each implementation of their system. This value is encrypted using the OEM's unique M_KEY and supplied to the data client as a userpermit. This method allows data clients to purchase licences to decrypt ENC cells.
SA	Scheme Administrator
SHA-1	Secure Hash Algorithm [3]
SSK	Self Signed Key (Self Signed Certificate File)
User Permit	Encrypted form of HW-ID uniquely identifying the ECDIS system

Chart Related Terms

ECDIS	Electronic Chart Display and Information System as defined by IMO
ENC	Electronic Navigational Chart as defined by the ENC Product Specification [1].
S-57	Transfer standard for ENC defined by IHO
SENC	System-ENC (This is the internal format that OEMs convert to when importing data)

Organisations

ECC	Electronic Chart Centre AS (www.ecc.as)
HO	Hydrographic Office (e.g. Data Server)
IHB	International Hydrographic Bureau
IHO	International Hydrographic Organisation
IMO	International Maritime Organisation
RENC	Regional ENC Coordinating Centre integrating ENCs from several HOs into a single service (e.g. Data Server)
UKHO	United Kingdom Hydrographic Office (www.ukho.gov.uk)

Computing Terms

CRC	Cyclic Redundancy Check
Dongle	Sometimes referred to as a hard lock device, It is a hardware device supplied by the OEMs that has the unique system identifier (HW_ID) stored securely within.
XOR	Exclusive OR

Page intentionally left blank

1 INTRODUCTION

The publication "S-63 IHO Data Protection Scheme", later referred to as 'the scheme', describes the recommended standard for the protection of ENC information. It defines security constructs and operating procedures that must be followed to ensure that the data protection scheme is operated correctly and to provide specifications that allow participants to build S-63 compliant systems and distribute data in a secure and commercially viable manner.

The Data Protection Scheme was prepared by the International Hydrographic Organisation's (IHO) Data Protection Scheme Advisory Group (DPSWG). The S-63 standard is based on the protection scheme developed and operated by Primar and Primar-Stavanger as part of their protected ENC service. The Electronic Chart Centre AS and United Kingdom Hydrographic Office were the original contributing organisations.

The Standard was adopted as the official IHO standard, by the IHO member states in December 2002 (IHO CL 66, 2002). It defines the roles and responsibilities for protecting ENC data produced by National Hydrographic Offices and distributed to customers with ECS/ECDIS systems.

1.1 General Description

This document specifies a method of securing ENC information and maintaining the integrity of an ENC service with multiple data services serving a large customer base. The purpose of data protection is threefold:

1. **Piracy Protection:** To prevent unauthorised use of data by encrypting the ENC information.
2. **Selective Access:** To restrict access to ENC information to only those cells that a customer has been licenced for.
3. **Authentication:** To provide assurance that the ENC data has come from approved sources

Piracy protection and selective access are achieved by encrypting the ENC information and providing cell permits to decrypt them. Data Servers will encrypt ENC data provided by producer nations before supplying it to the Data Client. The encrypted ENC is then decrypted by the ECS/ECDIS prior to being reformatted and imported into the systems SENC. Authentication is provided by means of digital signatures within the data.

The scheme does not specifically address how ENC or SENC information can be protected once it is within an end-user application. This is the responsibility of the OEMs.

The scheme allows for the mass distribution of encrypted ENCs on hard media (e.g. CD-ROM or DVD) and can be accessed and used by all customers with a valid licence containing a set of permits. Selective access to individual cells is supported by providing users with a licenced set of permits containing the encrypted cell keys. This licence is created using a unique hardware identifier of the system and is unique to each Data Client. Consequently licences cannot be exchanged between individual Data Clients.

The scheme uses a compression algorithm to reduce the size of the dataset. Unencrypted ENC data contains many repeating patterns of information, e.g. coordinate information. Compression is therefore always applied before the ENC information is encrypted and uncompressed after the decryption on the data client system (normally an ECS/ECDIS).

1.2 Participants in the Scheme

There are several types of users of the scheme, these are as follows:

- The Scheme Administrator (SA), of which there is only one.
- The Data Server (DS), of which there can be many.
- The Data Client (DC), of which there are many.
- The Original Equipment Manufacturer (OEM) of which there are many.

A more detailed explanation of these terms is given below.

1.2.1 Scheme Administrator

The Scheme Administrator (SA) is solely responsible for maintaining and coordinating the scheme. The SA role is operated by The International Hydrographic Bureau (IHB), as secretariat of the IHO, on behalf of the IHO member states.

The SA is responsible for controlling membership of the scheme and ensuring that all participants operate according to defined procedures. The SA maintains the top level digital certificate used to operate the S-63 Data Protection Scheme and is the only body that can certify the identity of the other participants of the scheme.

The SA is also the custodian of all documentation relating to the S-63 Data Protection Scheme.

1.2.2 Data Servers

Data Servers are responsible for the encrypting and signing ENC data in compliance with the procedures and processes defined in the scheme. Data Servers issue ENC licences (permits) so that Data Clients, with valid user permits, can decrypt ENC data.

Data Servers will use the M_KEY and HW_ID information, as supplied by the SA, to issue encrypted ENC cell keys to each specific installation. Even though the cell keys used to encrypt each cell are identical, they will be encrypted using the unique HW_ID and therefore cannot be transferred between other ECDIS from the same manufacturer.

Hydrographic Offices, Value Added Resellers and RENC organisations are examples of Data Servers.

1.2.3 Data Clients

Data Clients are the end users of ENC information and will receive protected information from the Data Servers. The Data Client's software application (OEM System) is responsible for authenticating the ENC digital signatures and decrypting the ENC information in compliance with the procedures defined in the scheme.

Navigators with ECDIS/ECS systems are examples of Data Clients.

The scheme does not impede agents or distributors from providing data services to their customers. Agreements and structures to achieve this are outside the scope of this document. This document contains only the technical specifications to produce S63 compliant data services and systems.

1.2.4 Original Equipment Manufacturers (OEM)

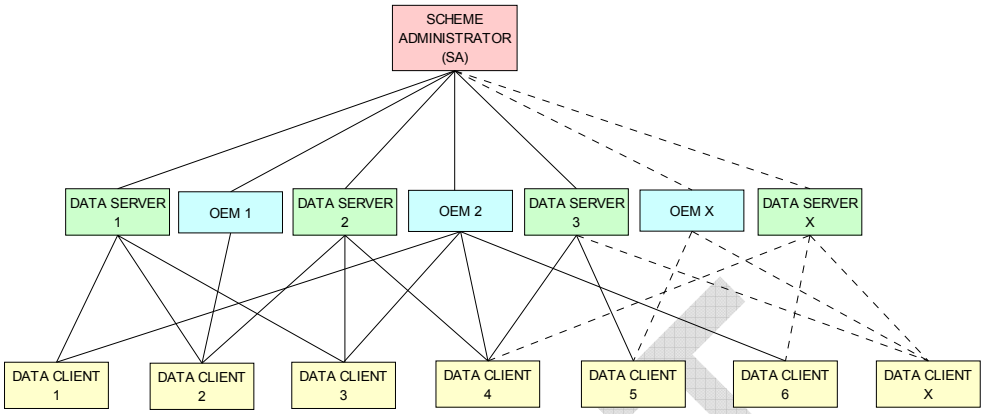
OEMs subscribing to the IHO S-63 DPS must build a software application according to the specifications set out in this document and self-verify and validate it according to the terms mandated by the SA. The S-63 standard contains test data for the verification and validation of OEM applications. The SA will provide successful OEM applicants with their own unique manufacturer key and identification (M_KEY and M_ID).

The manufacturer must provide a secure mechanism within their software systems for uniquely identifying each end user installation. The scheme requires each installation to have a unique hardware identifier (HW_ID).

The software application will be able to decrypt the cell keys using the HW_ID stored in either the hard lock or soft lock devices attached to or programmed within the application to subsequently decrypt and uncompress the ENC data. The CRC value contained within the ENC [1] can then be verified to establish the integrity of the underlying S57 data.

1.2.5 S-63 Participant Relationships

The Scheme Administrator (SA), of which there can only be one, authenticates the identity of the other participants within the scheme. All Data Servers and System Manufacturers (OEMs) must apply to the SA to become participants in the scheme and, on acceptance, are supplied with proprietary information unique to them. Data Clients are customers of Data Servers and OEMs where Data Servers supply data services and OEMs the equipment to decrypt and display these services.



IHO S-63 Data Protection Scheme Relationships

1.3 References

[1] S57 edition 3.1: IHO Transfer Standard for Digital Hydrographic Data, International Hydrographic Bureau (www.iho.shom.fr)
[2] Digital Signature Standard (DSS), FIPS Pub 186 (www.itl.nist.gov/div897/pubs/fip186.htm)
[3] Secure Hash Standard (SHA), FIPS Pub 180-1 (www.itl.nist.gov/div897/pubs/fip180-1.htm)
[4] Information Technology – Open Systems Interconnection – The Directory: Authentication Framework. X.509 version 3 - International Telecommunication Union
[6] ZIP File Format Specification, PKWare Inc.
[7] DES Modes of Operation, FIPS Pub 81 (www.itl.nist.gov/fipspubs/fip81.htm)
[8] RFC 1423: Privacy Enhancements for Internet Electronic Mail: Part III: Algorithms, Modes and Identifiers (ftp://ftp.isi.edu/in-notes/rfc1423.txt)
[9] Blowfish encryption algorithm, B. Schneier, Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994, pp. 191-204. (www.counterpane.com)
[10] CRC32 checksum algorithm. Information technology -- Telecommunications and information exchange between systems -- High-level data link control (HDLC) procedures. ISO/IEC 13239:2002.

1.4 Compatibility with Previous Versions

This version of S-63 uses the same algorithms and the same file formats and contents as the security scheme operated by Primar, Primar-Stavanger and IHO S-63 Version 1.0. This version of the S-63 standard has been amended to provide better definitions and explanation on the operation of the protection scheme.

A defined test data set has been produced and should be used by OEMs to verify and validate implementations of the S-63 Data Protection Scheme during self certification.

Version 1.1 of the standard has been produced in light of experience gained by Data Servers and ECS/ECDIS Manufacturers during the operation of the scheme under version 1.0. This version attempts to more clearly define the standard by removing duplication and possible ambiguity. It also contains additional mechanisms that will enable manufacturers to make their systems more intuitive for users of ECS/ECDIS. The following list refers to the revisions within the standard.

1. Removal of unnecessary duplication
2. Specification of how and under what conditions certain files must be used.
3. Removal of the permit dependency on the cell edition.
4. Additional information to enable Data Clients to manage ENC data more effectively and efficiently.
5. Identification of a loading strategy to enable more efficient loading of encrypted ENCs.

1.5 Document Structure

The main body of the document can generally be broken down into three parts. The first part details the components that are fundamental to the scheme and describes their purpose and construction. The second identifies how all the components come together within an S-63 ENC Exchange Set. The final part outlines the roles and responsibilities of each type of user participating in the scheme.

Main Document:

1. Scheme Components:
   - Section 2 Data Compression
   - Section 3 Data Encryption
   - Section 4 Data Licensing
   - Section 5 Data Authentication
   - Section 6 Data Management
2. Exchange Set Format and Structure
   - Section 7 Directory and File Structures
3. S-63 Participant Processes
   - Section 8 Scheme Administrators Processes
   - Section 9 Data Server Processes
   - Section 10 OEM & Data Client Processes

Additional Sections:

- S-63 Annex A: Data Server Certificate Request Procedure
- S-63 Annex B: Manufacturer Information Request Procedure

Appendices:

- Appendix 1: Contains a definition of available test data which can be used to develop full compliance with all aspects of the Data Protection Scheme.

1.6 Maintenance

Changes to this standard will conform to the *"Principles and a Set of procedures for making changes to IHO standards"*, as approved by the 13th CHRIS meeting (Athens, Sept. 2001).

1.7 Support

Support in using and implementing this standard is provided to users by members of the IHO DPSWG, via a security scheme discussion on the open ECDIS forum. (www.openecdis.org). In addition an inventory of frequently asked questions (FAQ) is maintained by the IHB on the ECDIS section of the IHO website (www.iho.shom.fr).

2 DATA COMPRESSION

2.1 Overview

An ENC file will, because of its structure, contain repeating patterns of information. Examples of this are the consecutive numbering of the feature object identifier (FOID) or small variations in the co-ordinate information within an ENC file. ENC data therefore responds well to compression with reductions in size of between 30% and 60% reducing greatly the cost of transfer of ENC data to its final destination. Only the ENC files (base and update) are compressed. The ENC files are always compressed before they are encrypted as the effectiveness of any compression algorithm relies on the existence of structured data contents.

2.2 Compression Algorithm

The security scheme uses the ZIP algorithm[1] [6] to compress and uncompress ENC data. It is identical to the algorithm used in many commercial applications e.g. WinZip, PKZIP. Potential Data Servers and OEMs should be aware that in the past errors have occurred when Data Servers compress data and it is interpreted by popular implementations of the ZIP algorithm as “text” data. If the data is uncompressed with incorrect parameters it can corrupt the ENC file leading to failing integrity checks. Data Servers and OEMs are advised to carefully implement compression/un-compression within their systems.

2.3 Compressed Files

The security scheme compresses only the ENC base cell and update files. No other files within the S-57 Exchange Set will be compressed.

[1] http://en.wikipedia.org/wiki/ZIP_%28file_format%29

Page intentionally left blank

3 DATA ENCRYPTION

3.1 What Data is encrypted?

Only one encryption algorithm is used within the Scheme. Only the data within the ENC Base or Update Cell files inside an S-57 Exchange Set, i.e. text or image files are left unencrypted. The scheme encrypts the complete content of the ENC Base or Update data files. Other information within the Scheme that is encrypted includes the OEM System HW_ID which is encrypted and provided to the Data Client in the form of a userpermit.

The cell keys used to encrypt the ENC data files are themselves encrypted by the Data Server and supplied to Data Clients as cell permits. Information about the encryption algorithm is available in section 3.2.3.

3.2 How is it encrypted?

Each single ENC cell file is encrypted using a unique Cell Key. The same Cell Key is used to encrypt all updates issued for the cell. The scheme however, allows for the cell keys to be incremented and changed at the discretion of the Data Server. The Cell Keys are delivered to Data Clients in the form of cell permits.

3.2.1 Encryption of ENC Information

The ENC information (base cells and updates) are encrypted using a 40-bit key.

3.2.2 Encryption of Other Protection Scheme Information

The Userpermit and the Cell permit contents are encrypted using a 48-bit key.

3.2.3 Encryption Algorithm – Blowfish

The scheme encrypts all information referenced in 3.1 using the Blowfish algorithm [9]. The algorithm is unpatented and available in the public domain (www.counterpane.com). Blowfish is a block cipher algorithm that operates on 64 bit (8 byte) quantities. It requires that the data sources must be padded if they are not a multiple of 8 bytes. The protection scheme uses the “DES in CBC Mode” padding algorithm defined in [8] whenever any data sources must be padded. This complies with the ECB (Electronic Code Book) mode of DES [7].

Page intentionally left blank

4 DATA LICENCING

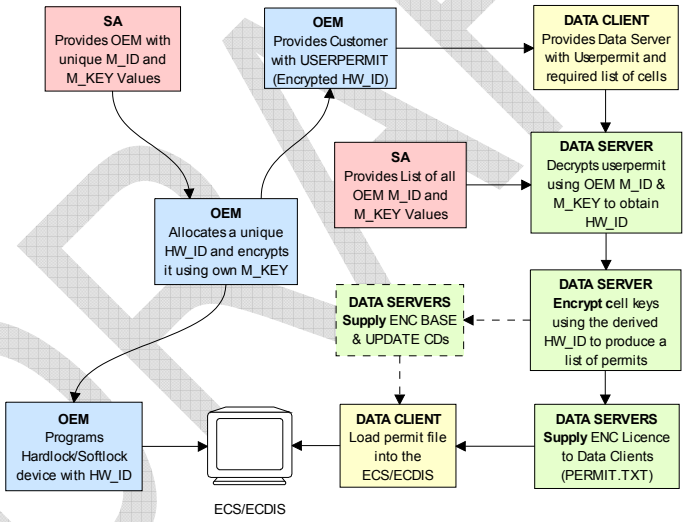
4.1 Introduction

Data Clients do not buy ENC data but are licenced to use it. Licensing is the method that Data Servers use to give Data Clients selective access to up-to-date ENC cells for a given period of time.

To operate the scheme effectively there must be a means where Data Client systems can unlock the encrypted ENC cells. To unlock the data the Data Clients system must have access to the cell keys that were used to encrypt the ENC cells. These keys are supplied to the Data Client, encrypted, in a permit file containing a set of cell permits. It is these cell permits that contain the encryption keys.

To make each set of cell permits exclusive the cell keys must be encrypted using something that is unique to the Data Clients system. OEMs assign a unique identifier (HW_ID) to each of their systems and provide an encrypted copy of this, in the form of a userpermit, to each Data Client. The HW_ID is stored in the userpermit encrypted.

OEMs encrypt the HW_ID with their own unique manufacturer key (M_KEY) so that a HW_ID cannot be duplicated by another manufacturer. Data Servers have access to the OEM M_KEYs and can therefore decrypt the HW_ID stored in the userpermit. Data Servers encrypt their cell keys with the manufacturers HW_ID when producing a set of cell permits. This makes them unique to the Data Client and as such not transferable between Data Client systems.



High Level ENC Licencing Diagram

4.2 The Userpermit

The userpermit is created by OEMs and supplied to Data Clients as part of their system so that they can obtain the necessary access to encrypted ENCs from Data Servers. The following section defines the composition and format of the userpermit.

All Data Clients with systems capable of using data, protected with the S-63 scheme, must have a unique hardware identification (HW_ID) built into their end-user system. Such a HW_ID is often implemented as a dongle or by other means ensuring a unique identification for each installation.

The HW_ID is unknown to the Data Client, but the OEM will provide a userpermit that is an encrypted version of the HW_ID and unique to the Data Client's system. The userpermit is created by taking the

assigned HW_ID and encrypting it with the manufacturer key (M_KEY). The CRC32 algorithm is run on the encrypted HW_ID and the result appended to it. Finally the manufacturer attaches their assigned manufacturer identifier (M_ID) to the end of the resultant string. The M_KEY and M_ID values are supplied by the SA and are unique to each manufacturer providing S-63 compliant systems.

The Data Client gains access to S-63 encrypted ENCs by supplying this userpermit to the Data Server who can then issue Cell Permits specific to it. Since the userpermit contains the manufacturers unique M_ID this can be used by Data Servers to identify which M_KEY to use to decrypt it. The M_ID is the last four characters of the Userpermit. A list of the manufacturer M_KEY and M_ID values is issued and updated by the SA to all Data Servers subscribing to the scheme. This list will be updated periodically as new OEMs join the scheme.

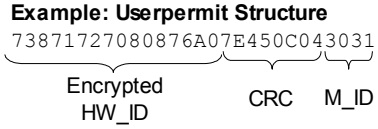
4.2.1 Definition of Userpermit

The userpermit is 28 characters long and shall be written as ASCII text with the following mandatory format and field lengths:

Encrypted HW_ID	Check Sum (CRC)	M_ID (Manufacturer ID)
16 hex characters	8 hex characters	4 hex characters

Any alphabetic character will be written in upper case.

Example: Userpermit Structure



4.2.2 HW_ID Format

The HW_ID is a 5 digit hexadecimal number defined by the OEM manufacturer. Such a HW_ID can be implemented as a dongle or by other means ensuring a unique identification of each installation[2]. The HW_ID must be stored within the system in a secure way.

The OEM manufacturer must assign a unique HW_ID for each installation. It is recommended that the HW_IDs are not sequential.

The HW_ID will be stored in an encrypted form in the Userpermit. It is encrypted using the Blowfish algorithm with M_KEY as the key resulting in a 16 digit (8 bytes) hexadecimal number. The encrypted HW_ID is then represented in its ASCII form in the userpermit as 16 characters.

Example of HW_ID is: **`A79AB`**
Example of encrypted HW_ID is: **`73871727080876A0`**

4.2.3 Check Sum (CRC) Format

The Check Sum is an 8 character hexadecimal number. It is generated by taking the encrypted HW_ID and converting it to a 16 character hexadecimal string. It is then hashed using the algorithm CRC32 [10] and the 4 bytes converted to an 8 character hexadecimal string.

The Check Sum is not encrypted and allows the integrity of the Userpermit to be checked.

The Check Sum in the above example is: **`7E450C04`**

[2] Manufactures, with the consent of the Data Server, may use the same HW_ID on more than one unit.

4.2.4 M_ID Format

The M_ID is a 2 digit hexadecimal number expressed as ASCII representation provided by the SA. The SA will provide all licenced manufacturers with their own unique Manufacturer Key and Identifier (M_KEY and M_ID) combination. The manufacturer must safeguard this information.

The SA will provide all licenced Data Servers with a full listing of all manufacturer codes as and when new manufacturers subscribe to the scheme. This information is used by the Data Server to determine which key (M_KEY) to use to decrypt the HW_ID in the Userpermit during the creation of Data Client cell permits.

The M_ID in the above example is: **`01`** or **`3031 (ASCII`**[3]**`)`**

4.2.5 M_KEY Format

The M_KEY is a 5 digit hexadecimal number provided by the SA. The OEM uses this key to encrypt assigned HW_ID when generating userpermits. The OEM must store it securely. This key is used by the Data Server to decrypt assigned HW_IDs.

Example of M_KEY is **`123AB`** or **`3132334142 (ASCII)`**

4.3 The Cell Permit

To decrypt an ENC cell the Data Client must have access to the encryption key (see section 3.2) used to encrypt it. Since the encryption keys are only known to the Data Server there needs to be a means of delivering this information to Data Clients in a protected manner. This information is supplied by the Data Server (e.g. RENC or VAR) to the Data Client in an encrypted form known as a cell permit. A single file is provided to deliver the cell permit and is named PERMIT.TXT (see section 4.3.1). This file may contain several cell permits based on the ENC coverage required by the Data Client.

The PERMIT.TXT file will be delivered either on hard media or using online services in accordance with the Data Servers operating procedures. These procedures will be made available to Data Clients when purchasing a licence.

Each cell permit record also contains additional fields that are supplied to assist OEM systems to manage the Data Clients licence and permit files from multiple Data Servers, see section 4.3.3.

Data Clients can obtain a licence to access ENCs by supplying the Data Server with their unique userpermit (see section 4.2). Data Servers can then extract the HW_ID from userpermit, using the Data Client's M_KEY, and create client specific cell permits based on this value. The format of a cell permit record is described below in sections 4.3.2 & 4.3.5.

Since Cell Permits are issued for a specific HW_ID they are consequently not transferable between installations (Data Client Systems). This method of linking the permit to the installation supports the production of generically encrypted CDs which can be distributed to all Data Clients subscribing to a service.

The Data Clients system decrypts the Cell Permit using the assigned HW_ID stored securely by hardware or software means. The decrypted cell keys can then be used by the system to decrypt the ENC cell. Since several Data Servers can make permit files for ENCs in their service, it is the responsibility of the Data Client system to manage permit files from several Data Servers.

4.3.1 The Permit File (PERMIT.TXT)

The Cell Permit will always be provided in a file called PERMIT.TXT and the file is completely encoded in ASCII[4] and contains 3 sections as follows:

[3] Note: The hex encoding may be unfamiliar to some readers. For historical reasons it has been preserved in this version of the standard. "1 2 3 4 5" is translated into "31 32 33 34 35" because the ASCII Base 16 representation of the character "1" is "31" etc. Though confusing at first this convention is used throughout the standard consistently as is standard hexadecimal and binary representations. To differentiate it is referred to as "(ASCII)"

[4] OEMs should be aware that all ASCII text files generated by the scheme may contain ambiguous end-of-line markers such as CR or CRLF and should be able to deal with these.

Section	Description
Header	This includes the file creation date and the format version.
:ENC	ENC permits (official) from the Data Server are listed under this section.
:ECS	ECS permits (non-official) from the Data server can be listed under this section.

The Data Server will make available information regarding how the permit files will be made available whether on hard media or online services. The following table defines the content and format of each section within the permit files separated by "new lines [NL]".

4.3.2 The Permit File - Header Formats

The following table defines the content and format of each section header within the permit file.

Section	Fieldname	Value
Date and time	**:DATE**	The field name, date and time is separated by a space character (SP <h20>). The date will be provided as **YYYYMMDD** and the time as **HH:MM** using the 24 hour clock. Example: **:DATE 20050809 11:11**
Meta Permit version	**:VERSION**	Integer in range 1 to 99. It will be incremented by 1 for each new version of the permit file format specification. S-63 Edition 1.1 defines the value as "2". i.e. **:VERSION** 2
Cell Permit type	**:ENC**	Field contains definition of permits available in an ENC distribution license from the Data Server. Field is identified with the following label in upper case **:ENC**
Cell Permit type	**:ECS**	Field contains definition of the meta permits available in an ECS distribution license from the Data Server. Field is identified with the following label in upper case **:ECS**

Example:
```
:DATE 20080809 11:11
:VERSION 2
:ENC
[List of licenced cell permits for official ENCs]
:ECS
[List of licence cell permits for other vector products]
```

4.3.3 Permit Record Fields

The Cell Permit Record is comprised of the following comma separated fields:

Field	Value
Cell Permit	As defined in section 4.3.4 & 4.3.5
Service Level Indicator	**0** for subscription permit **1** for single purchase permit
Edition Number	DSID-EDTN issue number of the ENC cell
Data Server ID	This is a two character alphanumeric issued by the SA
Comment	Free text field for comments on the cell permit etc.

4.3.4 Definition of the Cell Permit

The following table defines the fields contained in cell permit with a definition of the purpose of each.

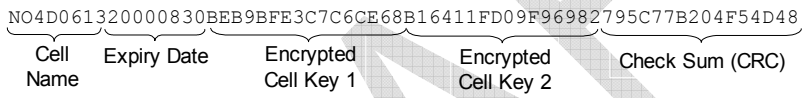
Field	Purpose
Cell Name:	The cell name enables Data Client systems to link the correct encryption key to the corresponding encrypted ENC cell file.
Expiry Date:	This is the date when the Data Clients licence expires. Systems must prevent any new ENC cells, new editions or updates created after this date from being installed.
Encrypted Cell Key 1 (ECK1)	ECK1 contains the decryption key for the current version of the ENC Cell.
Encrypted Cell Key 2 (ECK2)	ECK2 contains the decryption key to be used when the cell key is next iterated. The future key is contained within the cell permit to allow Data Servers to periodically change the Cell Key without simultaneously issuing new cell permits to all Data Clients.
Check Sum (CRC)	This value is provided to protect against tampering or accidental corruption.

4.3.5 Cell Permit Format

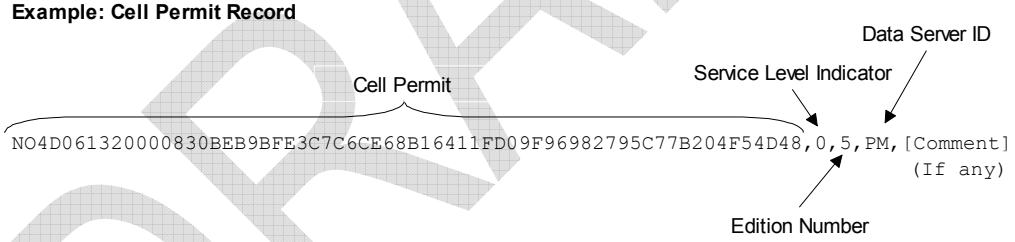
The Cell Permit shall be written as ASCII text with the following mandatory format and field lengths:

Field	Characters	Format
Cell Name	8	An alphanumeric string following the convention defined in S-57 Edition 3.1 Appendix B section 5.6 for cell names excluding the filename extension. Example is: **NO4D0613**
Expiry Date	8	A numeric string that contains the license expiry date for each ENC in the format **YYYYMMDD**. Example is: **20000830** (30th August 2000)
ECK1 & ECK2[5]	16	The Cell Keys are 5 byte random numbers – their hex representations are encrypted using Blowfish and then expressed in hexadecimal in the permit. **Note**: The blowfish encryption algorithm will cause the encrypted data to be padded to a multiple of 8 bytes in length. This means that encrypted Cell Keys are actually 8 bytes long, even though unencrypted they are only 5 bytes long (10 hex characters) Example: ECK1: **BEB9BFE3C7C6CE68** ECK2: **B16411FD09F96982**
ENC Permit Checksum	16	Contains the encrypted check sum for the Cell Permit. It is encrypted using the Blowfish algorithm with the Data Client's specific HW_ID and is an 8 byte number. This check sum is encrypted as opposed to the unencrypted check sum of the User Permit. e.g. The ENC Check Sum in the example below is: **795C77B204F54D48**

Example: Cell Permit Field



Example: Cell Permit Record



4.3.6 Additional Licence File (Optional)

Data Servers may wish to include an additional file with the PERMIT.TXT file to identify the licencee and provide information relating to system ID[6]. This file will be named **.LIC, where ** represents the data server ID.
Data client systems can access this file (if present) to display user information and provide userpermit information.

The file contains a single record with the following fields:

[5] The cell permit contains two fields for providing the data client system with the cell keys necessary to decrypt a specific ENC cell file. These fields may contain either two identical cell keys or two different cell keys and may differ between data servers. Some data servers may prefer to increment the cell keys only in the event of the security scheme is compromised others may prefer to periodically increment them according to their service procedures. The mechanism for data servers producing these keys is described in more detail in section 9.5.1. OEMs should note that any dependency on the edition number should be removed from theirs systems in edition 1.1 of the scheme.

[6] It may be useful when processing data client queries to have instant access to customer information such as licencing information and manufacturer ID. Data clients could supply this file with the query to speed up response times.

Field ID	Characters	Notes
Licensee	40	Name of company or individual signing the licence.
Vessel Name	40	Optional. This field may be left as spaces.
Fixed Site #1	240	Company Name and Address. This field contains free format text arranged in 6 40 byte sub-fields. Text will not cross the boundaries of the sub-fields.
Host System Name	40	For instance, Main, Backup, etc.
User Permit	28	Hexadecimal user permit
Licence Type	40	Service Indicator, e.g. Primar Stavanger ENC Service
HO data	36	Data for HO / Agent / Distributor use.

Total number of bytes: 464

5 DATA AUTHENTICATION

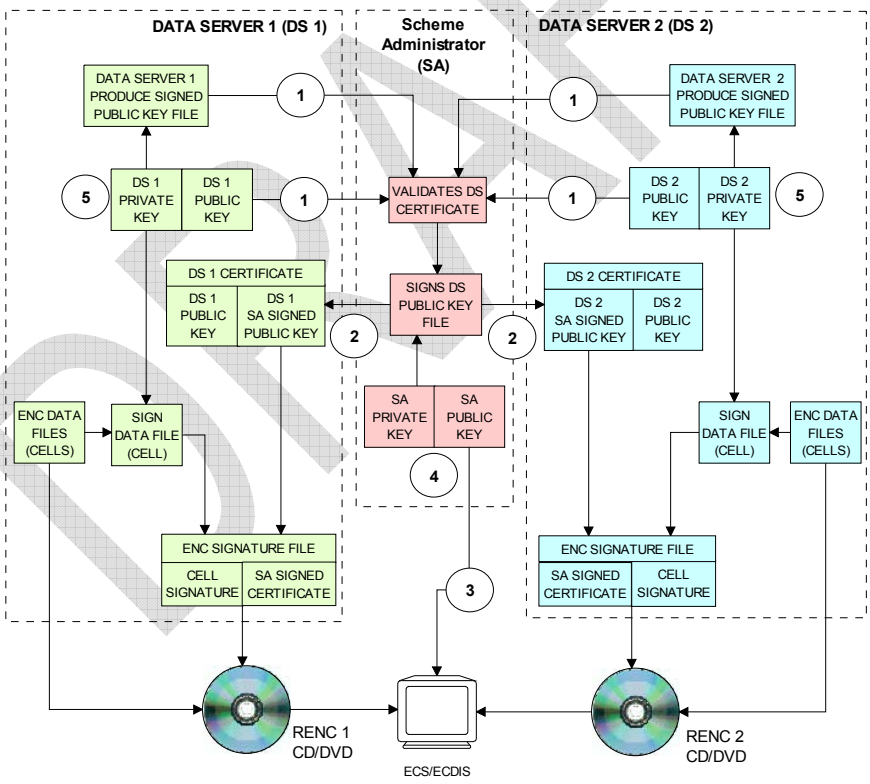
5.1 Introduction to Data Authentication and Integrity Checking

The digital signature technique used in the S-63 scheme uses a standard algorithm and key exchange mechanism widely used. S63 digital signatures use asymmetric public key algorithms within a PKI-like infrastructure scheme to unbreakably bind a data file with the identity of the issuer.

The scheme relies on asymmetric encryption[7] of a checksum of a data file. By verifying the signature against the issuer's public key, and also verifying the issuer's public key against a top level identity the user is assured of the signer's identity. A detailed explanation digital signatures is beyond the scope of this document and the reader is referred to the Digital Signature Standard (DSS), FIPS Pub 186 (www.itl.nist.gov/div897/pubs/fip186.htm) for a more detailed and accessible explanation.

The scheme can be considered to have three distinct phases:

1) A Scheme Administrator (SA) verifies the identity of a supplier of ENC information and provides the supplier with data to allow them to sign ENC data.
2) A Data Server (e.g. RENC or VAR) issues ENC data signed with their identity (and its verification by the SA).
3) The subsequent verification by the Data Client of the Data Server's identity (by its association with the SA) and the integrity of the ENC data.

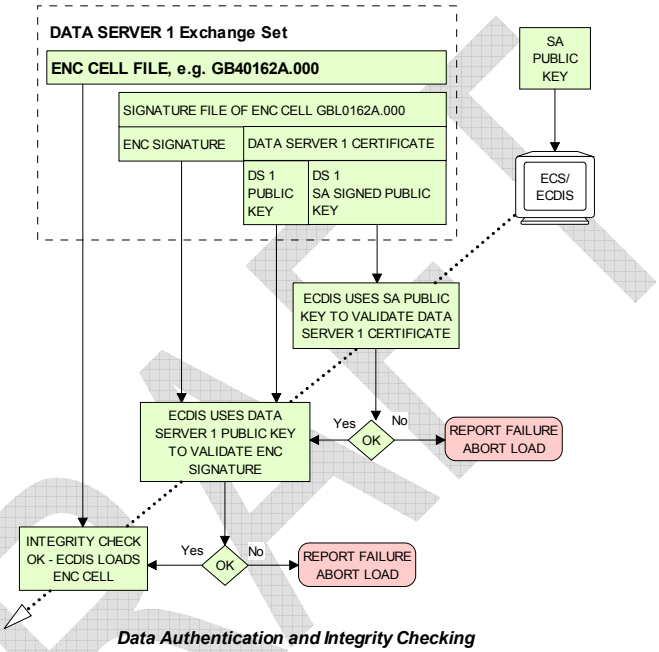


ENC AUTHENTICATION PROCESSES

[7] Asymmetric cryptography relies on algorithms where encryption and decryption take place with different cryptographic keys. Therefore one person can encrypt data and make available a decryption key for others to decrypt it. These keys are referred to as the "private key" and the "public key", collectively known as a "key pair"

NOTES – ENC AUTHENTICATION PROCESSES

1. The Data Server's Public Key and Self Signed Key (SSK) File are sent to the SA for validation when applying to join the IHO S-63 Data Protection Scheme.
2. If accepted the SA signs the Data Server's SSK with its own private key to produce a SA signed Data Server Certificate which is then returned to the Data Server.
3. The SA Public Key is widely distributed and installed independantly in OEM systems.
4. SA Public and Private Key pairs must be different from all other Data Servers.
5. All Data Server Public and Private Keys must be unique to each other and the SA.



Data Authentication and Integrity Checking

NOTES – DATA AUTHENTICATION AND INTEGRITY CHECKING

If an ECS/ECDIS is using the method depicted above, and if the SA Key Pair is different from the Data Server key pair, then it is able to authenticate and validate ENCs from Data Server 2 (or any other Data Server in the scheme) using the same SA public key.

1. **Authentication :** The ECS/ECDIS uses the SA public key, previously installed indepenantly of the CD, to check the certificate part of the signature file to confirm that the supplier's public key in the certificate is valid. That is, the Data Server is a bona fide member of the scheme
2. **Integrity Check:** The ECS/ECDIS uses the public key from the certificate to check the signature of the ENC cell (data) file.

5.1.1 SA Verification

The ECDIS needs to be able to verify that the ENCs are from a bona fide source. It does this by ensuring that the data server's public key provided within the ENC signature files can be validated against the SA's public key.

The SA provides certificates to each data server in the scheme; each certificate is unique, the SA only has to do this task once for each data server when they join the scheme. To obtain a certificate, data servers generate a key pair and provide the SA with their public key (as a self signed certificate); the SA (using their existing key pair) uses their private key to sign the data server's public key. The resulting certificate contains a signature of the supplier's public key. This certificate is then included within all ENC cells' and updates' signature files.

The SA makes their own public key widely known to the ECDIS community and OEMs should provide a means for the user to load this independently of the data.

5.1.2 Data Integrity

After the source of the ENC exchange set has been authenticated the ECDIS then checks data integrity by validating the signature file provided for each ENC by the data server.

The data server creates a signature file for each cell which consists of the following two parts:

- The signature of the dataset [which is created using the data server's private key, half of the data server key pair (in essence this is an encrypted checksum of the data) and is different for each cell]
- Their Data Server certificate (which remains constant).

The ECDIS uses the data server's public key that is included in the certificate to validate the data file signature (it decodes this data file signature and compares the checksum against the ENC cell). If this validation check is successful then it proves that the ENC has not been corrupted in any way and that the identity of the Data Server within the cell signatures is validated by the SA.

5.2 Digital Certificates (SA Authentication)

Certificates are digital files issued by a certification authority. They bind a specific public key together with other information to an individual or organisation. Certificates help prevent someone from using a fake public key to impersonate someone else. The scheme uses a chain of certificates, each one certifying the previous one until all parties are confident as to the identities in question. The SA certificate used by the IHO will be a self signed certificate[8] and is the **root certificate** for the scheme.

The SA will issue a digital certificate to all approved Data Servers by signing the Data Server's verified public key file. The following list of high level operations is performed in the issuing of digital certificates.

Scheme Creation

- SA creates a unique top level public and private key pair.

Establishment of a Data Server

- Data Server creates a unique public and private key pair.
- Data Server creates a Self Signed Key (SSK) by signing own public key file with own private key.
- Data Server supplies the SSK to the SA by a trusted means.
- SA verifies the Data Server's SSK using the Data Server's pubic key.
- SA signs the verified Data Server public key file using the SA private key.
- SA supplies the Data Server with its own unique SA signed Data Server Certificate.

Creation of Signed Data Sets

- Data Server verifies the resultant certificate with the SA public key (supplied separately).
- Data Server stores verified certificate and uses it in the creation of ENC signature files.

The format of the various files, certificates and signatures are described in more detail in section 5.4.

NOTE: the SA public key is made widely available to all interested parties, e.g. Data Servers, Data Clients and OEMs, in a number of ways, e.g. web, e-mail, etc.

5.2.1 The SA Public Key

The scheme requires that the SA public key is installed on the Data Client's systems independently of the ENC exchange set. This can be pre-installed by the OEM. However, the Data Client system must have a method of installing a new public key[9] on the system in the case where a new one is issued by the SA.

[8] The SA public key signed using the SA private key.

[9] It is envisaged that data servers will supply this independently of the exchange set to coincide with data that authenticates against the new public key.

If the user installs a new SA certificate or public key the system must confirm that a new one has been installed. If installing a new SA certificate (IHO.CRT the system must inform the user as follows:

"A new SA certificate (public key) has been installed this is valid to [enter expiry date] or unless the SA issues a new one for security reasons."

If installing a new SA public key (IHO.PUB) the system must inform the user as follows:

"A new SA public key has been installed this is valid until the SA routinely issues a new one or unless one is issued for security reasons."

Should the system report an authentication error during the loading process it should alert the user to the possibility that the SA may have changed the public key. Therefore a warning message must be displayed explaining the reason for this as follows:

"SSE 06 – The SA Certificate/Public Key is invalid. The SA may have issued a new public key or the ENC may originate from another service. A new SA public key can be obtained from the IHO website or from your distributor."

5.2.2 New Data Servers

The IHO, in conjunction with the DPSWG, will establish the identity of any organisation or commercial company wishing to join the protection scheme as a Data Server. If the SA revokes a Data Server Certificate, it will inform all Data Servers and Manufacturers about the change.

5.3 Digital Signatures (Verify Data Integrity)

A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and to ensure that the original content of the sent message is unchanged. Digital signatures are portable, easily verified and cannot be forged.

It is also acceptable for Hydrographic offices or other Data Server organisations (e.g. RENC/VAR) to use digital signatures to maintain provenance and data integrity between them in the delivery of ENC information. Each ENC file (both base and update files) will always have a single unique signature file associated with it. No other files in an encrypted ENC exchange set have a digital signature.

NOTE: An exchange set may contain signatures issued by different data servers and therefore each ENC file must be authenticated individually.

5.3.1 Technical Overview of Digital Signatures

Data authentication is provided using a digital signature compliant with the Digital Signature Standard (DSS) [2]. The DSS uses the Secure Hash Algorithm (SHA-1) [3] to create a message digest (hash). The message digest is then input to the Digital Signature Algorithm (DSA) [2] to generate the digital signature for the message using an asymmetric encryption algorithm and the 'private key' of a key pair. Asymmetric algorithms have the property that data encrypted using the 'private key' of the key pair can only be decrypted using the 'public key' of the key pair.

A consequence of encrypting the message digest with the private key is that anyone who has the public key (which as its name suggests can be made public) can decrypt and verify the message digest.
Further information on Digital Signatures and their use may be obtained from the IHO website (http://www.iho.shom.fr).

5.3.2 ENC Signature File Naming Convention

The digital signature file will match the cell file name except that the navigational purpose codes, digits 1–6, will be replaced by the characters I – N.

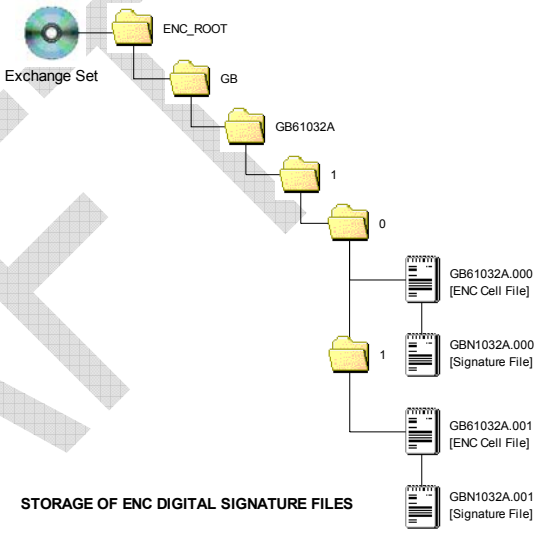
In general:
ENC file: `CC[1-6]XXXXX.EEE` (see S-57 Appendix B1)
Signature file: `CC[I-N]XXXXX.EEE`

Navigational Purpose	Signature Character
1. Overview	I
2. General	J
3. Coastal	K
4. Approaches	L
5. Harbour	M
6. Berthing	N

Example:
Cell file **GB100001.000** will have a signature file named **GBI00001.000**
Cell file **GB61032A.002** will have a signature file named **GBN1032A.002**

5.3.3 Storage of the ENC Signature File

The ENC signature file must be uniquely identifiable as belonging to a particular ENC data file as outlined in section 5.3.2 above. The digital signature file will always be located in the same directory as the ENC cell file that it relates to, as illustrated across.



STORAGE OF ENC DIGITAL SIGNATURE FILES

5.4 Data Authentication File Formats

There are a number of files associated with the authentication processes within the S-63 Data Protection Scheme. Among these are the certificate and signature files, as described in sections 5.2 & 5.3 and the private and public keys created to sign and authenticate them. Although these may be derived independently the various component parts contained within each file share common elements that are always formatted in the same way. The following table lists the files that are fundamental to the authentication of S-63 encrypted ENCs. This table also identifies those participants of the scheme who create them.

File Types	Scheme Administrator	Data Server
PQG File	✓	✓
Private Key (X file)	✓	✓
Public Key (Y file)	✓	✓
X509 v3 Certificate	✓	✘
Self Signed Key (SSK)	✘	✓
Certificate	✓	✘
Signature	✘	✓

5.4.1 File Elements

All elements comprise of two parts, a header and a data string. The following table lists all the possible elements that may go to make up a particular file, certificate or signature:

Element	Header	Data String
R	// Signature part R:	10 blocks of 4 characters.
S	// Signature part S:	10 blocks of 4 characters.
p	// BIG p	32 blocks of 4 characters.
q	// BIG q	10 blocks of 4 characters.
g	// BIG g	32 blocks of 4 characters.
x	// BIG x	10 blocks of 4 characters.
y	// BIG y	32 blocks of 4 characters.

5.4.1.1 Element Header and Data String Formatting

Each data string:

- Is preceded by a single header line. Header lines are indicated by two forward slashes (// ASCII - 0x2F2F) at the start followed by a space (SP ASCII 0x20) and the header characters in ASCII text as per the format descriptions below..
- Is expressed in ASCII text hexadecimal digits (0-9, A-F). Any alphabetic character will be in upper case.
- Is terminated by a full stop (. ASCII 0x2E).
- Has a space (ASCII SP 0x20) separating each group of 4 characters.
- Has a Carriage Return (ASCII CR 0x0D) and New Line (ASCII LF 0x0A) at the end of each data string.

5.4.2 Examples of File, Certificate and Signature Formats

The following section includes a set of examples of all the various files associated with this aspect of the S-63 Data Protection Scheme. A detailed explanation of how these files are created is outlined later in this document.

5.4.2.1 PQG Format

The PQG parameters are produced from a random string and are used in the creation of the X and Y private/public key pairs. After these have been made, the PQG parameters will be contained within the X and Y private/public key pairs.

P, Q and G are numerical parameters used in the Digital Signature Algorithm as input to the key creation process. Each data server can use a different set of P, Q and G or use an existing set to generate random key pairs. The Digital Signature Standard [2] describes their derivation and use.

Example of PQG Format:

```
// BIG p
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.
// BIG q
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.
// BIG g
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.
```

5.4.2.2 The X (Private Key) Format

The X file must be written as ASCII text in the following format:

```
// BIG p
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.
// BIG q
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.
// BIG g
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.
// BIG x
EBAF 2948 1485 7E7C 2F48 C7B2 9334 2F09 DA1A EB04.
```

5.4.2.3 The Y (IHO or Data Server Public Key) Format

Both the SA and Data Server public key are provided in the following format, the scheme uses a DSA Public Key of length 512 bits.

```
// BIG p
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.
// BIG q
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.
// BIG g
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.
// BIG y
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

5.4.2.4 The SA Digital Certificate (X509v3) Format

The SA Digital Certificate will be in X509v3 format [4] and represents a DSA Public Key of length 512 bits. The SA Digital Certificate will always be available in a file called IHO.CRT. The IHO.CRT file is available from IHO at http://www.iho.shom.fr.

All Data Servers providing an ENC service may include the SA certificate, for reference in the root directory of the media (e.g. in D:\IHO.CRT on a CD-ROM) but, as stated in Section 5.2.1, the installation on a Data Client's system of the SA certificate should be done independently. The check of the validity of the SA signature against each ENC signature must be done from the independently installed version of the SA certificate.

The SA public key in ASCII format (as opposed to the binary X509v3 format) is also made available on the IHO website at http://www.iho.shom.fr (the format is described in Section 5.4.2.3).

5.4.2.5 The Self Signed Key (SSK) Format

This is the file format that the Data Server uses to sign its own public key before sending to the SA for signing. The signature is the signature of the whole public key file (i.e. the PQG and Y parameters).

```
// Signature part R:
752A 8E5C 3AF5 6CCD 7395 B52E F672 E404 554F AAB6.
// Signature part S:
1756 E5C0 F4B6 BC90 4EC6 5F94 DF93 3ADF 68B8 86C4.
// BIG p
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.
// BIG q
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.
// BIG g
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.
// BIG y
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

Single Signature Element
Data Server (DS) Signature of DS Public Key

Data Server Public Key

Data Server Public Key File

The DS Signature is authenticated by the SA against the DS supplied Public Key

5.4.2.6 The SA Signed DS Certificate File Format

This is the file format used by the SA when it issues a Data Server Certificate file. The SA also uses a DSA Public Key of length 512 bits. The R & S pair is what is transcribed into the Data Server's ENC signature files.

```
// Signature part R:
8FD6 2AC7 27D2 8D0B CD27 BDF2 5CC6 9656 10E3 751F.
// Signature part S:
3DE7 DA37 5A40 80FC 4203 5C6E 37DE A984 2A88 2BDC.
// BIG p
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.
// BIG q
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.
// BIG g
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.
// BIG y
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

Single Signature Element
SA Signature of DS Public Key

Data Server Public Key

Data Server Public Key File

SA Signed Data Server Certificate

The SA Signature is authenticated by the SA Public Key held in the ECS/ECDIS

5.4.2.7 The ENC Signature File Format

The signature file must contain a signature and certificate pair. A file with just a signature is invalid as it does not certify the Data Server's identity. The ENC digital signature file has format, structure and order as in the following example:

```
// Signature part R:
77D3 4D86 DA6E 6E01 7058 7140 74FC 7E3D 21CD E80B.
// Signature part S:
04A1 7B52 081F B6CE 10FE 5AD9 1CCE 3F25 FEAC DA05.
// Signature part R:
8FD6 2AC7 27D2 8D0B CD27 BDF2 5CC6 9656 10E3 751F.
// Signature part S:
3DE7 DA37 5A40 80FC 4203 5C6E 37DE A984 2A88 2BDC.
// BIG p
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.
// BIG q
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.
// BIG g
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.
// BIG y
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

First Signature Element
Data Server Signature of ENC Data File

Second Signature Element
SA Signature of the Data Server Certificate

Data Server Public Key

Data Server Public Key File

SA Signed Data Server Certificate

The first Signature/R & S Pair is authenticated by the Data Server's Public Key

The second Signature/R & S Pair is authenticated by the SA Public Key stored in the ECS/ECDIS

The second R and S pair is used to authenticate the Data Server digital certificate (p, q, g and y strings). If verified successfully, the Data Server public key (y string) can be extracted and used to verify the digital signature (first R and S pair) of the encrypted ENC. This allows the Data Client to verify the SA digital certificate, to extract the Data Server public key, and to verify the digital signature of the ENC data.

6 DATA MANAGEMENT

6.1 Introduction

The loading and import of ENCs to an ECS/ECDIS must be carefully managed; this is especially true in a multiple Data Server environment. Since the scheme encrypts the entire contents of an ENC cell file (base and update), this restricts access to certain subfields in an ENC cell file required by OEM systems to manage the import of ENCs to the ECS/ECDIS SENC. As a consequence additional S-63 files are necessary to supplement this inaccessible data as well as modification to an existing S-57 file, i.e. the CATALOG.031 file.

It has also been identified that the import of large numbers of ENCs has rendered some aspects of S-57 impractical to implement, e.g. a single exchange set split across multiple media volumes. For this reason it has been necessary to modify the loading strategy and utilise the additional S-63 files to better manage the installation and loading of ENCs across multiple exchange sets.

As mentioned previously S-63 is designed to operate in a multiple data supplier environment. S-57 does not have a mechanism to discriminate between ENC exchange sets supplied by different data servers and therefore it has been necessary to cater for this in version 1.1 of this standard.

The additional S-63 files contain important information that, if used correctly, can make the S-57 import process more efficient and intuitive for the Data Client. These are outlined in more detail below in this section.

The loading/import method can be broken down into the following processes:

- Manage the import of Data Server specific ENC exchange sets using the Data Server ID.
- Manage a Data Server's service if extended across multiple exchange sets.
- Manage the import of licenced ENC cells in a contiguous manner, ensuring all ENC base cells and corresponding update files, if any, are imported correctly and sequentially.
- Manage the import of text and picture files by maintaining a relationship between them and the cell file they are associated with.

The following table lists the additional S-63 files and file modifications together with their main purpose within S-63. These files and their associated formats are described in more detail at section 6.2, 6.3 & 6.4.

File/Field	Primary Management Function
PRODUCTS.TXT	Required to provide the following information: 1. The Issue Date of the Products File being installed. 2. A catalogue of all available cells in a Data Server's service. 3. The coverage of all available cells in a service. 4. The latest Issue Date of all available cells in a service (including cancelled cells). 5. The destination exchange set (base or update) where the ENC base cell file (EN Application Profile) resides. This may be a base cell, new edition or re-issue.
SERIAL.ENC	Required to provide the following information: 1. The Data Server ID. 2. The Exchange Set week number and date of issue. 3. Type of Exchange Set (base or update) 4. The Exchange Set number in a multiple series.
CATALOG.031 [CATD-COMT]	Required to provide information originally contained in the DSID field of the cell file to import the complete, sequential content of an exchange set.

6.2 ENC Product Listing (PRODUCTS.TXT)

The file named 'PRODUCTS.TXT' will be supplied with each encrypted exchange set and will be stored in a folder named 'INFO' held in root directory. It is the mechanism for managing data within an ENC Service and exchanging it with data already held in the Data Client's system SENC. The structure and format of this file is described in more detail in sections 6.2.1, 6.2.2, 6.2.3 & 6.2.4.

There are two types of PRODUCTS.TXT file, “PARTIAL” and “FULL”. A partial products listing contains the current status of all ENCs contained in a single exchange set. A full product listing contains the current status of ALL cells in a Data Server’s service, that is, all exchange sets. Although procedures may vary between Data Servers a full product listing will always be provided with the weekly update exchange set.

NOTE: OEMs should ensure that their systems are able to handle “FULL” and “PARTIAL” product listings (section 6.2.2 refers). Base CDs may contain a partial listing with only the contents of the CD included. The Update will always carry a full product listing of all available ENCs in a Data Server’s Service.

Licence and ENC information from different Data Server’s must be stored independently on the manufacturer’s system. The SERIAL.ENC file (see section 6.3) contains the Data Server ID and should be used in conjunction with the associated product listing to identify the source of the service. The latest product listing contains the current status of ENC cell data in a service. This file is used to compare available ENC cell data in the exchange set with information already stored in the OEMs SENC. The OEM system can then determine what new data is available for import.

It is recommended that OEMs maintain a copy of the latest product listing on their systems to reflect the current status of a particular service. To manage both “FULL” and “PARTIAL” product listings it is essential that new information is merged with existing stored data and not overwritten.

6.2.1 Product List File Structure

The content of the product list will be divided into sections. The Product List file is completely encoded in ASCII and contains 3 sections as follows:

Section	Description
Header	Contains general information about the nature of the Product List, e.g. the time of creation, version number.
:ENC	Contains the current status of all ENC cells/updates provided by the data server.
:ECS	Contains information about other digital chart information provided by the data server.

6.2.2 Product List Header

The Product List will always start with one Header Section. The Header Section will consist of several records. Each record will start at a new line and be terminated with ASCII CR/LF characters as within the signature files.

The Header will consist of the information fields defined in the example and table below. All the fields are mandatory and will always be defined in the same order.

Field	Fieldname	Value
Date and Time	:DATE	YYYYMMDD HH:MM The field name, date and time will be separated by a <space> character. The date will be provided as 20060627 and the time as 09:00:00 using the 24 hour clock. Example: **:DATE 20061019 09:00:00**
Product List Version	:VERSION	Integer in range 1 to 99. It will be incremented by 1 for each new version of the PRODUCTS.TXT file specification. S-63 Edition 1.1 defines the value as “2”. i.e. **:VERSION 2**
Content	:CONTENT	“FULL” Full copy of Product List “PARTIAL” Partial copy of Product List Code used to indicate if the Product List file contains a full or partial copy of the complete Product List. Example: **:CONTENT FULL**

Example:

```
:DATE 20061019 09:00:00
:VERSION 1
:CONTENT FULL
```

6.2.3 Product List 'ENC' Section

The Product List will always contain one ENC Section. It will contain information about the current navigational status of all official ENC cells and updates supported by the Data Server.

This section will start with one *ENC Section Identifier* record as defined below.

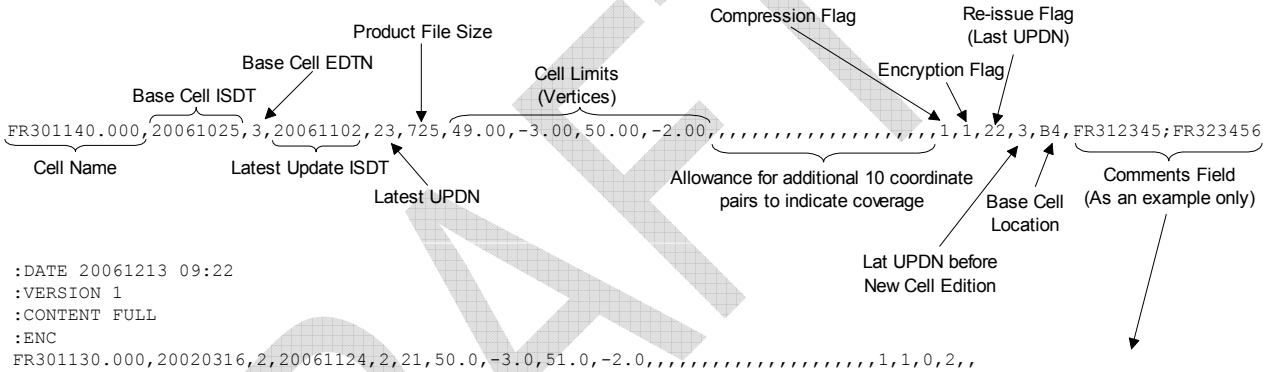
Field	Fieldname	Value
ENC Section Identifier	:ENC	Not applicable

The ENC Section will then consist of repeating records defining the status of each ENC supported by the data server. The definition of this record is defined in the table below:

Field	Value
Product Name	Name of product as defined in S57e3 DSID-DSNM subfield. The file extension will always be 000. Example: **GB202400.000**
Base Cell Issue Date [EN Application Profile]	YYYYMMDD This date is only used for the base cell files (i.e. new data sets, re-issue and new edition), not update cell files. All updates dated on or before this date must have been applied by the producer. Example: **20050222**
Base Cell Edition	Edition number of base [EN] ENC cell. Integer in range 1 to 999 Identical to content of S57e3 DSID-EDTN. In the case where a cell is cancelled the Product Edition will be set to 0. This allows the ECDIS system to quickly identify cells that have been removed from a service.
Issue Date Latest Update [ER Application Profile]	YYYYMMDD Date on which the latest update for the current ENC cell edition was issued. This field is used whenever there is an update or a re-issue of the cell.
Latest Update Number	Integer in range 1 to 999 Update number of the latest update message issued for the ENC cell edition. Identical to content of DSID-UPDN. Left blank when no update is available for the current edition of the base cell. Used only for updates and re-issues.
File Size	Integer in range 1 to 999999 Total file size in Kilobytes for all files issued for the product. This will include the size for the base cell, updates and any applicable text and picture files.
Cell limit Southernmost latitude	Degrees of arc, south is negative. Southernmost latitude of data coverage in the ENC product. Example: **49.898773299986 (49°53´.93N)**
Cell limit Westernmost longitude	Degrees of arc, west is negative. Westernmost longitude of data coverage in the ENC product. Example: **-1.927277300003 (001°55´.64W)**
Cell limit Northernmost latitude	Degrees of arc, south is negative. Northernmost latitude of data coverage in the ENC product. Example: **50.922828000014 (50°55´.37N)**
Cell limit Easternmost longitude	Degrees of arc, west is negative. Easternmost longitude of data coverage in the ENC product. Example: **-0.000166700008 (000°00´.01W)**
10 Data Coverage Coordinates	Optional. Degrees of arc, south and west are negative. 10 coordinate pairs can be supplied to indicate the data coverage within the ENC cell. It will be provided as repeating Y-coordinate and X-coordinate pairs.
Compression	Integer in range 0 to 99 "0" No compression "1" Compression is used (see section 2)
Encryption	Integer in range 0 to 99 "0" No encryption "1" Encryption is used (see section 3)
Base cell update number	In the event of a cell being re-issued the update number current at the time of the re-issue should be inserted here. If a cell edition does not have a re-issue then this field is blank or zero filled.
Last update number for previous edition	Empty if no previous editions available in the data server database. If previous editions of the cell are available then this field will contain the last update number for the previous edition.

Base Cell Location	**CD-ROMs** The location within the exchange set where the base cell can be found. Base cells may be located on either one or several base or update exchange sets. This is an integer in range 1 to 99 proceeded either by a 'B' if on a base CD or 'U' if on the update, e.g. **B7**, **B11**, **U1**, etc. **Large Media Support** In the case where a service supports large media this field is divided into two subfields delimited by a "**;**" (semi colon). The first subfield contains the media number ID and the second the exchange set number. The Media ID is designated with a **"M"** followed by a number. The ExSet number is formatted in the same as for CD-ROMs, e.g. **"B1"**. For example a base cell could be located in the following ways, **"M1;B1"**, **"M1;B2"**, **"M2;B10** ,etc. Updates for example, **"M1;U1"** or **M1;U2** if more than one update ExSet on the same media. See **Appendix 2** of this document for details.
Cancelled Cell Replacements (Old Comments Field))	If a cell is cancelled and a replacement cell(s) is issued this field is used to identify the replacement(s). In cases where there are more than one replacement the cell names will be delimited by a "**;**" (semi-colon).

Example of Structure and Format:




```
:DATE 20061213 09:22
:VERSION 1
:CONTENT FULL
:ENC
FR301130.000,20020316,2,20061124,2,21,50.0,-3.0,51.0,-2.0,,,,,,,,,,,,,,,,,,,,1,1,0,2,,
FR301140.000,20061025,3,20061102,23,725,49.0,-3.0,50.0,-2.0,,,,,,,,,,,,,,,,,,,,1,1,22,3,FR312345;FR323456,
FR301150.000,20030412,3,20061124,26,744,48.0,-3.0,49.0,-2.0,,,,,,,,,,,,,,,,,,,,1,1,0,8,,
FR301160.000,20020413,2,20051206,13,87,49.0,-4.0,50.0,-3.0,,,,,,,,,,,,,,,,,,,,1,1,0,5,,
FR301170.000,20020803,2,20061003,12,502,48.0,-4.0,49.0,-3.0,,,,,,,,,,,,,,,,,,,,1,1,0,3,,
FR301180.000,20030830,3,20061114,1,34,49.0,-5.0,50.0,-4.0,,,,,,,,,,,,,,,,,,,,1,1,0,3,,
```

6.2.4 Product List 'ECS' Section

The Data Server may also issue other types of digital chart products such as backdrop charts that can be used to display chart coverage. Information about these products can also be made available in the Product List if the data server wishes to.

The content of this section is identical to the ENC Section defined in 6.2.3. The only difference is the *Section Identifier* which will be "**:ECS**".

Encoding example of a Product List which utilises all the features defined in section 6.2.1.

```
:DATE 20061019 09:00:00
:VERSION 1
:CONTENT FULL
:ENC
AR201130.000,20051118,1,20060703,1,,-36.43335487,-57.41667361,-34.69998565,-54.33335853,,,,,,,,,,,,,,,,,,,,1,1,0,0,,
AR302120.000,20051219,1,20060427,2,,-39.44997766,-62.39166614,-38.74168723,-61.11683505,,,,,,,,,,,,,,,,,,,,1,1,0,0,,
AR402490.000,20051206,1,20060330,1,,-39.11668811,-61.94017540,-38.95167627,-61.76656919,,,,,,,,,,,,,,,,,,,,1,1,0,0,,
AR402550.000,20051219,1,20060427,1,,-39.01664968,-62.16649373,-38.88332240,-61.94017540,,,,,,,,,,,,,,,,,,,,1,1,0,0,,
AR402560.000,20051122,1,20060427,1,,-38.99166872,-62.39166614,-38.74168723,-62.16649373,,,,,,,,,,,,,,,,,,,,1,1,0,0,,
AR420010.000,20060912,2,,,,-35.16832135,-56.07497834,-35.03499407,-55.84166992,,,,,,,,,,,,,,,,,,,,1,1,0,3,,
:ECS
PM1WORLD.000,19990101,1,,,3000,-90,-180.0,90.0,180.0,-90.0,-180.0,90.0,180.0,,,,,,,,,,,,,,,,,,,,0,0,,
```

6.3 Serial File (SERIAL.ENC)

A file named SERIAL.ENC is supplied so that Data Clients can identify the following information prior to import:

- Data Server ID (registered with the SA)
- Week of Issue
- Date of Issue
- CD Type (Base or Update)
- Format Version
- Exchange Set Number (of a series of exchange sets)

6.3.1 SERIAL.ENC File Format

The SERIAL.ENC file is provided to assist Data Client's systems manage the import ENC CDs supplied by a specific Data Servers across multiple exchange sets. It should be the first file that is read from the exchange set as it contains important information about the IHO assigned Data Server's ID, the CD Publication Date, type of CD, number of CDs in that particular Data Server's service, etc.

The contents of this file can be cross referenced with the installed permits to check the status of the Data Client's subscription status.

Field ID	Domain	Bytes	Range	Notes (see below)
Data Server ID	character	2	Any two alphanumeric	1
Week of Issue	character	10	Any ASCII characters	2
Date of publication	date	8	YYYYMMDD	3
CD Type	character	10	BASE or UPDATE	4
Format version	decimal		01.00 – 99.99	5
Volume ID	character	6	V01-99X01-99	6
End of record delimiter	hexadecimal	3	0x0B0D0A	7

Notes	Explanation and Description
1	Data Server ID should be registered with the IHO; where the data server is also an HO the Agency code for the organisation is obtained from S-62 – IHO Codes for Producing Agencies.
2	The week of issue specifies the week and year that the CD is distributed, e.g., **`WK12-99`**, **`WK45-99`**, **`WK23-00`**, etc.
3	Date of publication is in the regular date format, **`YYYYMMDD`**, e.g., **`19990414`**, **`20000102`**, **`20061102`**, etc.
4	The CD can be issued in two different types: BASE: The format should be defined as BASE, the CD contains all ENs and any additional ERs. UPDATE: The format should be defined as UPDATE, contains any new ENs and all ERs issued since the issue of the last relevant Base CD.
5	Format version describes the version of the SERIAL.ENC file. The present version is 01.00
6	This field may be used to show the exchange set number of a series, e.g. **`V02X03`**. That is, exchange set 2 of a series of 3.
7	The end of the record delimiter consists of binary characters, and therefore care should be taken when attempting to edit the file – it cannot be edited in Windows Notepad! This is the reason why the SERIAL.ENC file must always be edited in an ASCII/Hexadecimal editor. The delimiter does not normally need to be changed. The delimiter used is **`0x0B0D0A`**.

The SERIAL.ENC file should be stored directly under the media root file, i.e. on the same level as the ENC_ROOT and INFO directories.

Example of SERIAL.ENC file

```
PRWK15-99 19990414UPDATE01.000V02X03x0B0D0A
```
(Where `x0B0D0A` is the end of record delimeter converted to hex)

6.4 The S-57 Catalogue File (CATALOG.031)

The "Data Set Identification" [DSID] field is used by ECS/ECDIS to ensure that base cells and update files are imported to the SENC in the correct sequence and without omission. Since the complete ENC Cell file is

encrypted, information in the DSID field of each cell file is not available to OEM systems, unless it is decrypted first.

The “Comments” [CATD-COMT] field in each cell record of the CATALOG.031 file is used to store the required DSID information. Since the CATALOG.031 file acts as the table of contents for the exchange set and identifies where all files are stored it is ideally suited for this purpose.

The information stored in this field must be identical to that stored in the DSID field of the cell file which in turn must conform to section 5.7 of the IHO S-57, Appendix A, Product Specification. This is summarised in the table below. This table specifies the rules for encoding ENC EN & ER application profiles.

Event	File Extension	EDTN	UPDN	UADT	ISDT	Comments
New ENC Cell	.000	1	0	19950104	19950104	UADT = ISDT
Update 1	.001	1	1	Prohibited	19950121	ISDT only
Update 2	.002	1	2	Prohibited	19950225	ISDT only
...						
Update 31	.031	1	31	Prohibited	19950905	ISDT only
Re-issue of an ENC Cell	.000	1	31	19950905	19950910	UADT < or = ISDT
Update 32	.032	1	32	Prohibited	19951023	ISDT only
...						
Update 45	.045	1	45	Prohibited	19951112	ISDT only
New edition of ENC Cell	.000	2	0	19951201	19951201	UADT = ISDT
Update 1 to edition 2	.001	2	1	Prohibited	19960429	ISDT only
...						

Data Servers must extract the necessary information from the DSID field prior to encryption and encode it in the CATD-COMT of the CATALOG.031 file. The structure and format of this field is described in more detail in section 6.4.1. Data Client systems must then read the CATD-COMT field as though accessing the DSID field in an unencrypted exchange set.

The issue date of an ENC base cell or update file is always encoded in the DSID-ISDT sub-field of the ENC data set files. The DSID-ISDT field that is copied to the CATD-COMT field enables the OEM systems to quickly check if the ENC information has been issued after the licence expiration date without first having to decrypt the ENC.

6.4.1 The CATD-COMT Structure and Format

The DSID information stored in the CATD-COMT field is subdivided into four or five comma separated subfields. This is dependant on whether the ENC file has an EN or ER application profile. The final subfield is punctuated by a semi colon (;).

Examples:

```
VERSION=1.0,EDTN=1,UPDN=0,UADT=20060703,ISDT=20060703;
VERSION=1.0,EDTN=1,UPDN=1,ISDT=20060710;
```

6.4.1.1 Cancelled Cells

The following definition is taken from the IHO S-57 Product Specification and details how a cell should be deleted (cancelled):

“In order to delete a data set, an update cell file is created, containing only the Data Set General Information record with the ‘Data Set Identifier’ [DSID] field. The ‘Edition Number’ [EDTN] subfield must be set to 0. This message is only used to cancel a base cell file.”

Since an update is provided containing just the delete message it should be treated as an ER. Therefore for the purposes of the CATD-COMT field it should be encoded as follows:

```
VERSION=1.0,EDTN=0,UPDN=2,ISDT=20060814;
```

The following table illustrates the conditions that apply to all the different types of transactions.

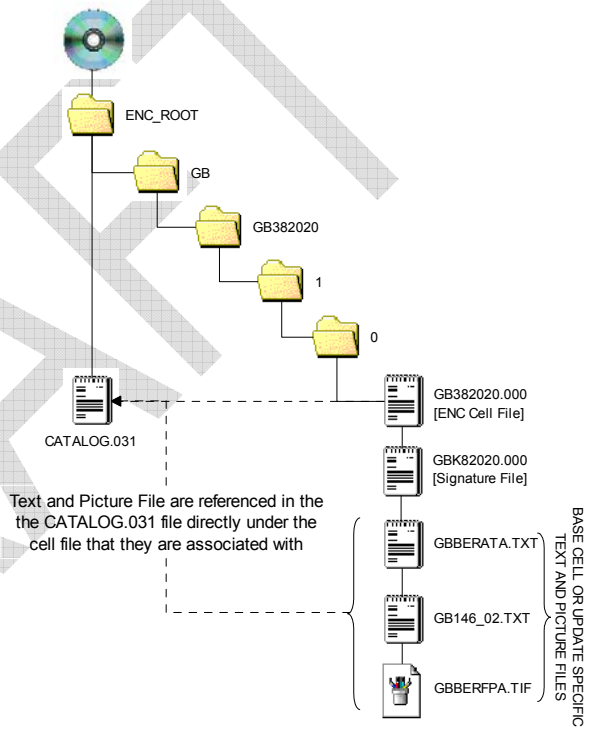
Version Number	Edition Number	Update Number	Update Application Date [UADT]	Issue Date [ISDT]	Comment
VERSION=1.0	EDTN=1	UPDN=0	UADT=20060703	ISDT=20060703	New Cell (EN)
VERSION=1.0	EDTN=1	UPDN=1	Prohibited	ISDT=20060710	Update (ER)
VERSION=1.0	EDTN=1	UPDN=10	UADT=20060710	ISDT=20060717	Re-issue (EN)
VERSION=1.0	EDTN=1	UPDN=11	Prohibited	ISDT=20060724	Update (ER)
VERSION=1.0	EDTN=2	UPDN=0	UADT=20060731	ISDT=20060731	New Edition (EN)
VERSION=1.0	EDTN=2	UPDN=1	Prohibited	ISDT=20060807	Update (ER)
VERSION=1.0	EDTN=0	UPDN=2	Prohibited	ISDT=20060814	Cancelled Cell (ER)

6.4.2 Text and Picture File Management

OEM systems can identify and manage text and picture files in an unencrypted ENC file by reading through it to see if there are any correctly formatted references. However, for reasons mentioned previously this is not possible in an encrypted ENC file, short of decrypting it first. To get around this problem Data Servers must identify and manage these files before the data is encrypted.

Data Server systems must read all unencrypted ENC files in their service and associate any text and picture files with the corresponding ENC base cell or update file. Any associated text and/or picture files should be recorded in the CATALOG.031 directly beneath the ENC cell file (base or update) record they relate to.

NOTE: Multiple copies of the same text files may be present in the exchange set that are associated with more than one ENC data file.



Example of Text (TXT) and Picture (TIF) File Storage

Example (partial CATALOG records):

```
GB\GB382020\1\0\GB382020.000 (ENC Cell File)
GB\GB382020\1\0\GBBERATA.TXT (Text File)
GB\GB382020\1\0\GB146_02.TXT (Text File)
GB\GB382020\1\0\GBBERFPA.TIF (Picture File)
GB\GB382020\1\0\GBK82020.000 (ENC Signature File)
```

OEM systems can then manage these files when reading the CATALOG.031 file during import.

Page intentionally left blank

7 DIRECTORY and FILE STRUCTURE

7.1 Introduction

The scheme does not mandate the use of a particular directory or file structure. However, because the entire ENC data file is encrypted it is difficult to maintain an association of text and picture files with the relevant ENC cell file (base or update). The directory structure that has been adopted by existing Data Servers is given in an example at the end of this section at 0. This structure enables text and picture files to be safely managed as described in section 6.4.2

7.2 S-57 File Management

The directory structure is not mandated and may vary between Data Servers. The location of all S-57 files in an encrypted exchange set is defined in the CATALOG.031 file. That is, the path to all files in the exchange set is specified in each file record.

7.3 File Format

As well as the exchange set [ENC_ROOT] the root directory contains a folder named ”INFO” that contains the PRODUCTS.TXT file and any additional, or extra, yet to be defined, files as specified by individual Data Servers. The root directory also contains the SERIAL.ENC file. Each Cell file in the exchange set under ENC_ROOT has a corresponding signature file.

OEMs should note that the IHO.CRT file will only be supplied in the exchange set for a limited period in support of legacy systems, after which it will be withdrawn.

7.4 Folder and File Naming

All folders and files must be named according to the conventions set out in the IHO S-57 Product Specification and this document. All folders and files in an S-63 encrypted exchange set must be in UPPER CASE.

7.5 Exchange Set Media

Data Servers may supply exchange sets to Data Client’s using several different methods, for example:

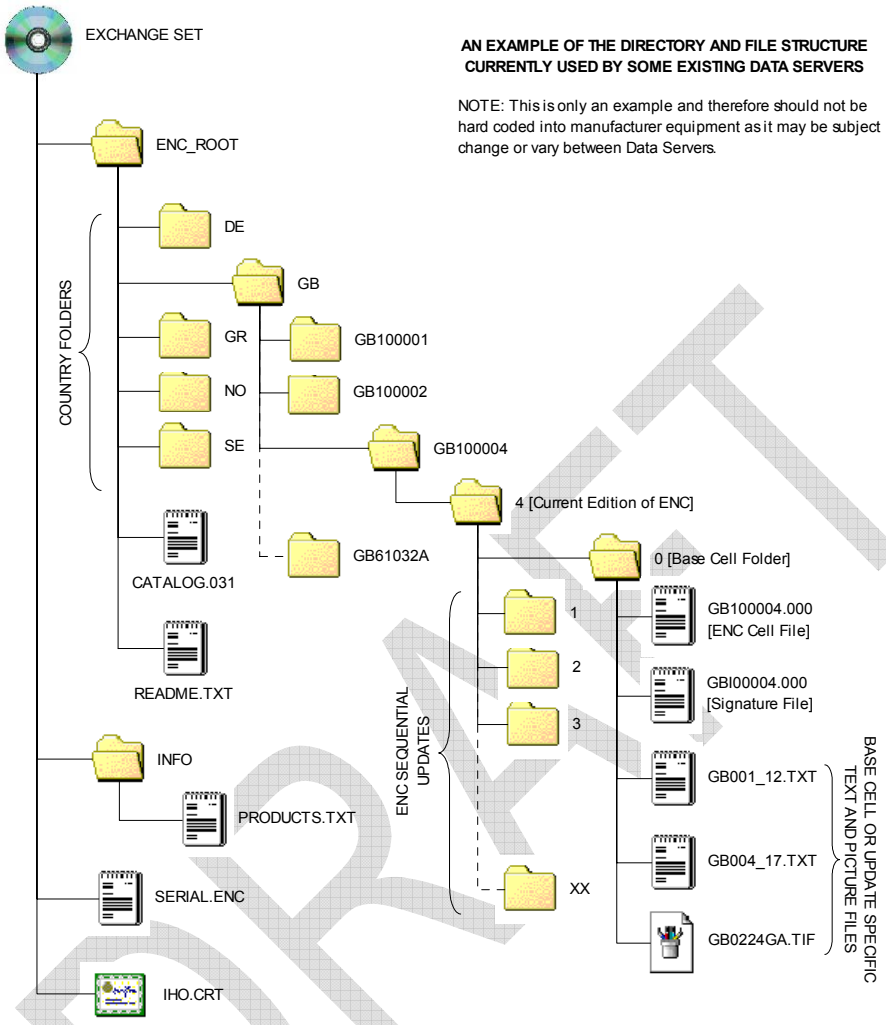
1. CD-ROM
2. Large Media Support
3. On-line Services

7.5.1 CD-ROM

Encrypted exchange sets delivered by this method will be given the volume ID consistent with the IHO S-57 Data Protection Scheme. Example of S-63 Directory Structure

7.5.1.1 Folder Definitions

Although the example below is based on a Base CD the Update CD is very similar, the only difference being that the update does not necessarily hold all the base cell data. However the update must contain data that is consistent and sequential for the Base CD to which it applies.



NOTE: The location of all files in the exchange set can be obtained using the CATALOG.031 file.

7.5.2 Large Media Support

Large Media Support is defined as devices capable of storing much larger volumes of data than a standard CD-ROM. Details relating to the storage of S-63 encrypted ENCs on such devices is given at Appendix 2.

7.5.3 On-Line Services

Data Clients can download exchange sets from RENC/VAR as defined by the service provider. The download is then copied to a hard media and depending on the media the RENC/VAR will advise on the volume ID to assign to the media.

NOTE: OEMs can program their systems to automatically detect an exchange set or a group of exchange sets however; this should not be hard coded into the system. If the media contains an unexpected format the system should default to a browse facility so that users can manually specify the location of the root directory of a required exchange set.

8 SCHEME ADMINISTRATOR PROCESSES

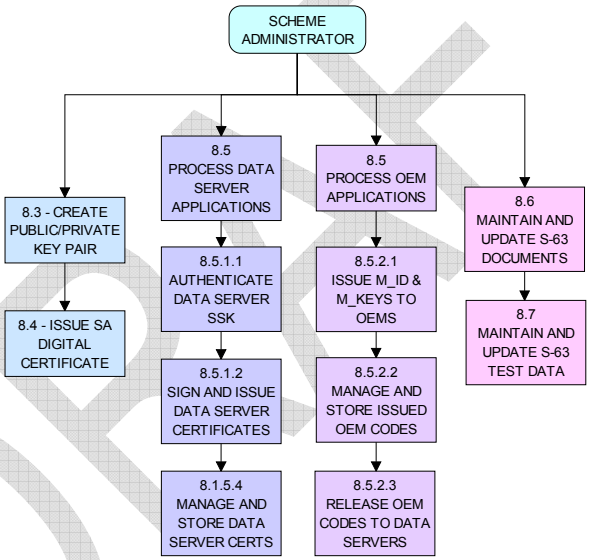
8.1 Data Protection Scheme Administrator

The Data Protection Scheme Administrator (SA) is solely responsible for maintaining and coordinating the S-63 Data Protection Scheme (DPS). The SA role is operated by The International Hydrographic Bureau (IHB), as secretariat of the IHO, on behalf of the IHO member states.

The SA is responsible for controlling membership of the scheme and ensuring that all participants operate according to defined procedures. The SA maintains top level encryption keys used to operate the complete scheme and is the only body that can issue certificates to other participants. The SA is the custodian of all documentation relating to the scheme.

8.2 Scheme Administrator Processes

The main responsibilities of the IHO as the S-63 Scheme Administrator are depicted in the following diagram. Each *"Process Box"* cross references the particular section where these operations are outlined in more detail.



Main Scheme Administrator Processes

8.3 Create Top Level Key Pair

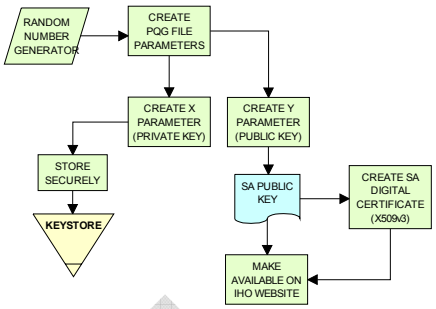
The IHO as scheme administrator must create a top level public and private key pair. The private key will be used to sign Data Server certificates and the public key to authenticate the signature. The public key must be installed on the Data Clients system independently of the Encrypted ENC data.

8.3.1 Create PQG Parameters

This procedure is normally performed by the SA and Data Servers during the creation of public/private key pairs. Although the PQG parameters generated by Data Servers do not need to be identical to those contained within the SA public key and the SA Digital Certificate, the key lengths used must be identical.

The PQG file only exists by itself for a short period during the creation of the X and Y files. After these have been made, the PQG file will be contained within the X and Y files.

The creation of appropriate PQG parameters is covered in more detail in the Digital Signature Standard (DSS) [2]. For information relating to the format of the PQG file see section 5.4.2.1.



Make Top Level Key Pair

8.3.2 Create Private Key

The private key is an output of the key generation process. The private key must be stored securely and access limited to only those persons that have a need to know. Unauthorized possession of the SA's private key can potentially undermine the security of the authentication part of the scheme. The SA will issue a new public key (and corresponding SA certificate) if the private key is compromised. Details of the X file (Private Key) format are contained in section 5.4.2.2.

8.3.3 Create Public Key

The public key is an output of the key generation process. The public key is sent to all participants of the scheme in both a digital and a printed form. The two forms are to be sent by different means. Details of the Y file (Public Key) format are contained in section 5.4.2.3.

8.4 Create and Issue SA Digital Certificate (X509v3)

The SA Digital Certificate will be compliant with X509v3 [4]. The SA Digital Certificate will always be provided in a file called IHO.CRT. The IHO.CRT file is available from IHO at http://www.iho.shom.fr.

The SA uses a DSA Public Key of length 512 bits.

All Data Servers providing an ENC service may include the SA certificate, for reference in the root directory of the media (e.g. in D:\IHO.CRT on a CD-ROM) but, as stated in Section 6.1, the installation on a Data Client's system of the SA certificate should be done independently. The check of the validity of the SA signature within each ENC signature file must be done from the independently installed version of the SA certificate.

The SA public key (as opposed to the digital certificate) is also made available as an ASCII file on the IHO website at http://www.iho.shom.fr (the format is described in Section 6.5).

8.4.1 Update SA X509v3 Digital Certificate (Public Key)

The SA will publish and make a new SA Digital Certificate available under the following circumstances:

- When the SA Digital Certificate expires. In this case the Certificate shall not contain a changed public key.
- When the SA private key has been compromised. In this case a new public key shall be contained within the SA Digital Certificate.

The SA will publish its new Digital Certificate and, if applicable, a new printable version (ref section 6.5) of the public key on the IHO website (http://www.iho.shom.fr). All Data Servers and Manufacturers will immediately be informed and will receive copies of the new Digital Certificate and, if applicable, the new public key in printable format.

The Data Server and Manufacturers are collectively responsible for informing their Data Clients of any new SA Digital Certificate and, if applicable, any new SA public key.

This procedure is normally performed by all users of the protection scheme when a new SA Digital Certificate or public key is issued and is performed as follows:

- Obtain the new SA Digital Certificate and printable SA public key from the IHO website (http://www.iho.shom.fr)
- The application shall load the new SA Digital Certificate and check the public key and the printable public key are identical. Only when this has been done is the application to assume that the SA public key is correct. This same process is applied to the replacement of the original SA public key.
- Replace the existing SA Digital Certificate with the newly issued certificate.

8.5 Process Applications from Data Servers & OEMs

The Scheme Administrator is responsible for processing applications from Data Servers and OEMs who wish to subscribe to the IHO S-63 ENC Data Protection Scheme. This includes the management and issuing SA Signed certificates to Data Servers and the management and issuing of Manufacturer Codes (M_ID & M_KEY) to accepted OEMs and their distribution to authorised Data Servers. The application processes are outlined in more detail at **ANNEX A** & **ANNEX B** of this document.

8.5.1 Process Data Server Request for Data Server Certificate

A successful Data Server applicant will be required to supply a certificate signed with the Data Server private key, this is known as Self Signed Key (SSK). The certification process is then carried out by the SA and is detailed step by step below.

8.5.1.1 Authenticate Self Signed Key (SSK) File

The SA authenticates the Data Server SSK file before creating and issuing a Data Server Certificate. Initially the SA should confirm that the SSK has been supplied in the correct format as described in section 5.4.2.5, if incorrect the process should be terminated and a warning given. If correct the process to authenticate should proceed as follows:

a) Extract the signature elements 'R' and 'S' (i.e. the first two data strings and their attendant headers from the SSK file supplied by the Data Server). This leaves a public key file.
b) Hash the public key file using the algorithm SHA-1. All bytes within the file are to be hashed.
c) Verify the signature (the elements removed at 'a' above) by passing it, together with the public key file and the hash of the public key file (as obtained at 'b' above) to the DSA. This will return a status (correct or incorrect).

If the signature verifies correctly, the SA can produce the Data Server Certificate.

8.5.1.2 Create Data Server Certificate

The SA creates a Data Server Certificate after the SSK has been authenticated. The details of the digital signature algorithm DSA are covered in the publication FIPS 186 Digital Signature Standard (DSS). The procedure is as follows:

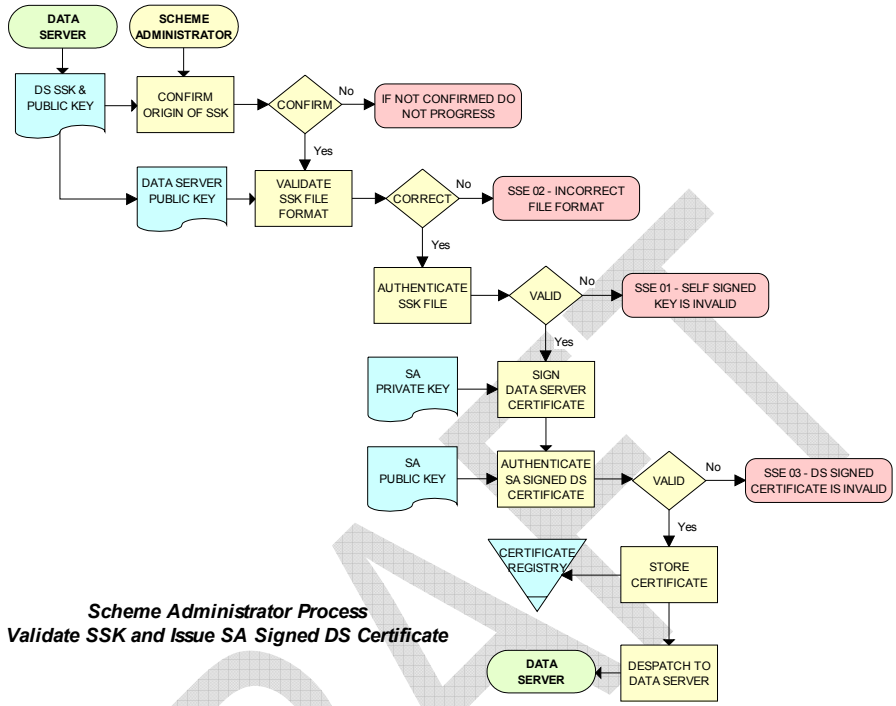
a) Discard the signature elements (i.e. the first two data strings and their attendant headers) from the self signed key file. This leaves a public key file.
b) Hash the public key file using the algorithm SHA-1. All bytes within the file are to be hashed.
c) Sign the public key file (as hashed at 'b' above) by passing the SA private key, the hash of the public key file (as obtained at 'b' above) and a random string to the DSA. This will return the two signature elements ('R' and 'S').
d) Write these to the certificate file and append the public key file (as left at 'a' above) to form the certificate.

8.5.1.3 Authenticate SA signed Data Server Certificate

The SA confirms the newly signed certificate is valid before despatching it to the Data Server. The procedure is as follows:

a) Extract the signature elements (i.e. the first two data strings and their attendant headers) from the newly created DS certificate file. This leaves the DS's public key file.
b) Hash the DS public key file (obtained from 'a') using the algorithm SHA-1. All bytes within the file are to be hashed.
c) Verify the signature elements (as removed at 'a' above) by passing it, together with the SA public key and the hash of the DS public key file (as obtained at 'b' above) to the DSA. This will return a status (correct or incorrect).

If the DS Certificate authenticates correctly, it can be sent to the DS and used in the construction of ENC digital signatures.



Scheme Administrator Process
Validate SSK and Issue SA Signed DS Certificate

8.5.1.4 Manage Data Server Certificates

When a new SA signed Data Server Certificate has been issued to a Data Server it should be stored securely in a certificate store. The certificate should be uniquely assigned to the Data Server and cross referenced to the private key used to sign it and the public key used to confirm authentication.

8.5.2 Process OEM Application

Manufacturers must apply to the SA to become a member of the IHO S-63 Data Protection Scheme.

8.5.2.1 Issue and Manage S-63 Manufacturer Codes

Successful OEM applicants will be supplied with their own unique Manufacturer ID (M_ID) and Manufacturer Key (M_KEY) see section 0 and 4.2.5. These codes must be stored securely together with the manufacturers contact details and whether they are still an active participant in the scheme.

8.5.2.2 Issue M_ID and M_KEY listings to Data Servers

Data Servers require the M_ID and M_KEY values so that they can identify a specific manufacturer and derive the correct M_KEY for extracting the Data Clients HW_ID from the userpermit. The SA will supply Data Servers with a complete list of codes for all approved manufacturers of S-63 compliant systems. This list will be supplied in a protected form every time a manufacturer is added to the list or if the status of a manufacturer changes, e.g. membership of the scheme revoked.

8.6 S-63 Test Data

The S-63 data protection scheme is supported by a comprehensive set of test data, see S-63 Appendix 1 - Data Protection Scheme Test Data.

8.7 *Scheme Administrator – Security QA Procedures*

8.7.1 Documentation

The SA shall hold the documentation for the Data Protection Scheme. This shall be held under change control procedure and the SA shall inform all participants (Data Servers and developers of Data Client applications) of the Data Protection Scheme, of changes to the standard.

Test data for the Data Protection Scheme and a software kernel are also available for system manufacturers to test their implementation for full compliance. The test data and software kernel are described in Appendix A and B and obtainable from the IHO website (http://www.iho.shom.fr).

8.7.2 Administration of Confidentiality Agreement

All details required to operate the security scheme and all proprietary information (e.g. M_KEY) will be provided to interested parties under cover of a Confidentiality Agreement. The SA shall be responsible for administering this agreement. The Confidentiality Agreement will limit the possibilities for participants to breach the Data Protection Scheme.

8.7.3 Audit of Security Registers

The SA shall have the ability to audit all security registers maintained by the participants of the Data Protection Scheme. The content of these registers are defined in Sections 9.3.2.3, 9.3.3.3, 9.7.3 and 10.9.3. The SA shall audit these registers to confirm that they are complete and up-to-date. Any problems must be corrected immediately or the participant shall become non-compliant and optionally may be withdrawn from the protection scheme.

8.7.4 Creation of M_IDs and M_KEYs

The SA shall be responsible for creating and issuing the M_ID and M_KEY values used within the Data Protection Scheme. The SA shall record, in a M_ID / M_KEY Register all M_ID/M_KEY values and which organisations have received which values. The SA will ensure that no duplicate values are created.

The SA will provide information to all Data Servers in the protection scheme on amendments to M_ID and M_KEY values.

8.7.5 Creation of Digital Signature Keys (Private and Public Keys)

The SA shall have the ability to create a private and public key pair. The private key is used in the certificate signing process and the public key in the signature authentication process.

The private key must be stored securely and access to it limited to only those persons that have a “need to know”. The SA will issue a new public key (and corresponding SA certificate) if the current private key is compromised.

The SA public key should be made available to all participants of the S-63 Data Protection Scheme in both digital and printed forms, e.g. fax and downloadable from a website. The two formats are to be sent or made available by different methods.

8.7.6 Acceptance of Self Signed Keys (SSK)

The SA shall confirm that any self signed key provided by a Data Server is bona-fide by contacting the originating organisation. This can be done either by phone, fax or mail but the origins must be confirmed to the Scheme Administrator’s satisfaction before the DS certificate is signed by the SA using the self signed key. The SA is to record all SSKs received in a SSK Register.

8.7.7 Creation of Data Server (DS) Certificates

The SA shall be able to create SA signed DS Certificates from the self signed keys provided by a DS and the SA private key. The signed certificate should be authenticated against the DS public key before being sent to the DS. The SA shall keep a record of all DS certificates in a DS Certificate Register.

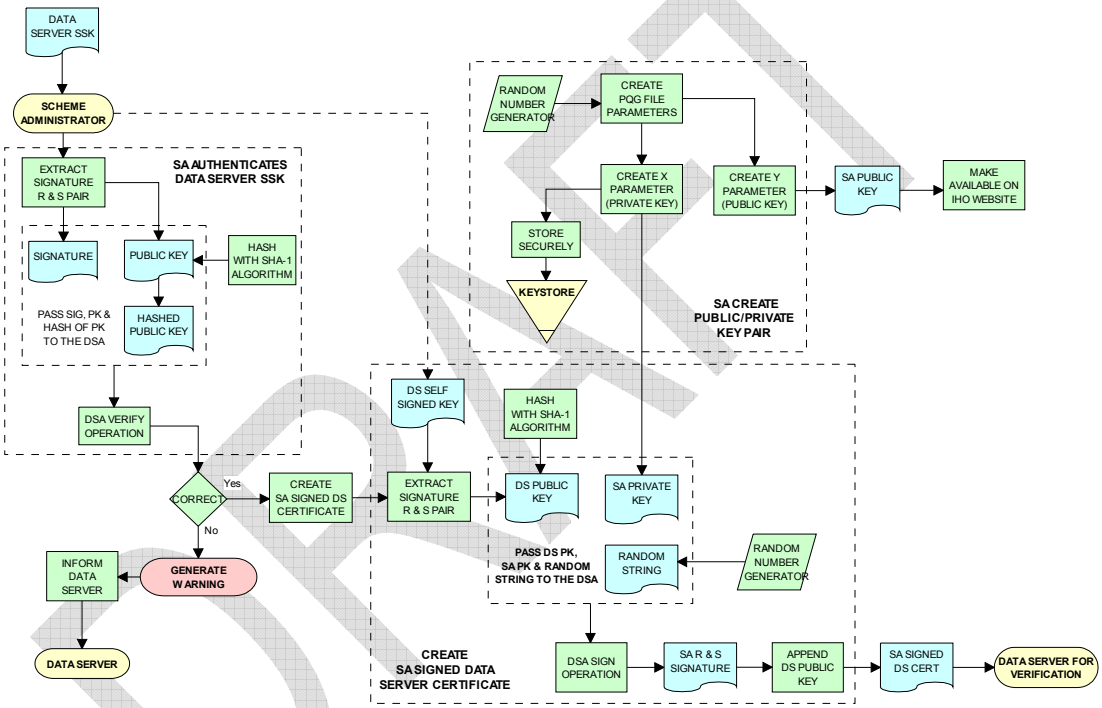
The DS will be required to sign a Confidentiality Agreement before the SA will issue the DS Certificate. The SA will provide information to all participants of the protection scheme on any revoked Data Server Certificates.

8.7.8 Creation of Random Strings

In order to sign data (required as part of the certificate creation), the SA will have to create random strings. The SA shall ensure that the same value is not used for any two separate signings. Although it is not possible to guarantee this if the strings are generated randomly. However, the chance of the same string being generated twice is extremely small.

8.7.9 Handover of M_ID and M_KEY

When a system manufacturer completes their internal compliance testing, they will be required to sign a Confidentiality Agreement before the SA will issue the M_ID and M_KEY.



Scheme Administrator (SA) - SSK Authentication & Certificate Signing Process

9 DATA SERVER PROCESSES

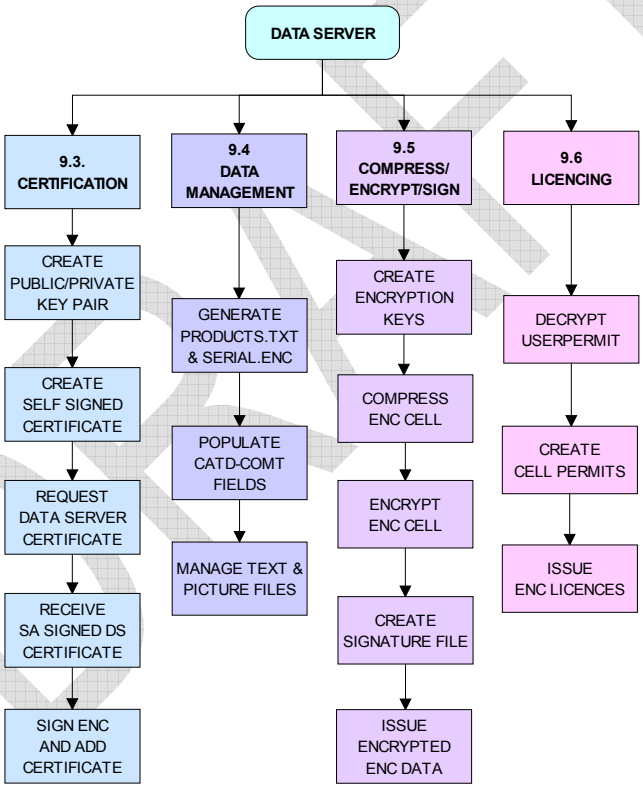
9.1 Overview

Data Servers are responsible for encrypting and signing the ENC information in compliance with the procedures and methods defined by the IHO S-63 Data Protection Scheme.

Hydrographic Offices and RENC organisations are examples of Data Servers. Organisations wishing to become a Data Server must first sign and submit an ***S-63 Data Server Agreement*** *together with a completed* ***Data Server Certificate Request Form***. This process is outlined in more detail on the IHO website (www.iho.shom.fr).

9.2 Data Server Processes

The main responsibilities of approved Data Servers operating under the IHO S-63 Data Protection Scheme are depicted in the following diagram. Each top level *"Process Box"* cross references the particular section where these operations are outlined in more detail.



Main Data Server Processes

9.3 Certification Processes

9.3.1 Produce Public/Private Key Pair

Data Servers will need to create a Private and Public Key pair as part of the 'asymmetric' encryption methodology adopted by the IHO S-63 Data Protection Scheme. The Data Server's Public and Private Keys are used in the following functions:

- The Private Key is used to sign the Data Server's Public Key to create a Self Signed Certificate (SSK).
- The Public Key is used to validate the SSK before it is supplied to the SA.
- The Private Key is used to sign all compressed and encrypted ENC data files produced by the Data Server.
- The Public Key is used to check the integrity of the ENC data files in the ECS/ECDIS.

9.3.1.1 Create PQG Signature Parameters

This procedure is normally performed by the SA and Data Servers during the creation of public/private key ***pairs***. Although the PQG parameters generated by Data Servers do not need to be identical to those contained within the SA public key and the SA Digital Certificate, the key lengths used must be identical.

The PQG file only exists by itself for a short period during the creation of the X and Y files. After these have been made, the PQG file will be contained within the X and Y files

The creation of appropriate PQG parameters is covered in the publication Digital Signature Standard (DSS) [2].

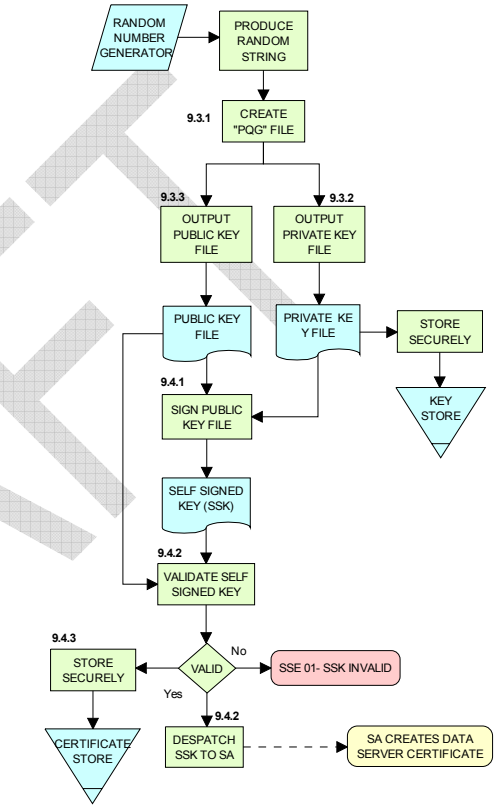
9.3.1.2 Create Private Key File

The private key is an output of the PQG generating process. The private key must be stored securely and access limited to only those persons that have a need to know.

Unauthorized possession of the Data Server's private key can potentially undermine the security of the authentication part of the scheme. The Data Server will issue a new public key if the private key is compromised. Details of the X file (Private Key) format are contained in section 5.4.2.2.

9.3.1.3 Create Public Key File

The public key is an output of the PQG generating process. The public key is contained in the SA signed Data Server Certificate that forms part of the ENC Signature File (see section 5.4.2.7). The Data Clients system extracts the public key element of this file to check the integrity of the ENC data file against the ENC signature. Details of the Y file (Public Key) format are contained in section 5.4.2.3.



Data Server Processes
Create Public/Private Key Pair
Produce & Authenticate Self Signed Key (SSK)

9.3.2 Create Data Server Self Signed Key (SSK)

The SSK is created and submitted to the SA to obtain a Data Server Certificate. The SSK contains the Data Server public key with a signature created by the Data Server. Although the SSK format is identical to the Data Server Certificate defined in section 8.5.1.2, the only difference is the SSK is created by the Data Server and the Data Server Certificate which is created and issued by the SA.

The SSK defines a signature of the Data Server's public key. The input to the signature should be the Data Server's public key, formatted according to the Public Key file format as described in Section 5.4.2.3. The SSK file shall be written as ASCII text with the format, structure and order described in section 5.4.2.5.

9.3.2.1 Sign Public Key and Generate SSK

This procedure is normally performed once by a Data Server to create its self signed key (SSK) which is then sent to the SA who will use it to create a Data Server Certificate. The details of the digital signature algorithm *DSA* are covered in the publication FIPS 186 Digital Signature Standard (DSS) [2]. The procedure is as follows:

a) Hash the public key file using the algorithm *SHA-1* [3]. All bytes within the file are to be hashed.
b) Sign the public key file (as hashed at 'b' above) by passing the private key file, the hash of the public key file (as obtained at 'b' above) and a random string through the *DSA* algorithm [2]. This will return the two signature elements ('R' and 'S').
c) Write these to the self signed key file in the format defined in Section 5.4.2.5 and append the public key file to form the self signed key file.

9.3.2.2 Authenticate/Validate Data Server SSK

Data Servers must authenticate the SSK against the Data Server public key to confirm that a valid SSK has been produced.

a) Extract the signature elements 'R' and 'S' (i.e. the first two data strings and their attendant headers from the SSK file supplied by the Data Server). This leaves a public key file.
b) Hash the public key file using the algorithm SHA-1. All bytes within the file are to be hashed.
c) Verify the signature (the elements removed at 'a' above) by passing it, together with the public key file and the hash of the public key file (as obtained at 'b' above) to the DSA. This will return a status valid or invalid.

If the SSK is valid then it can be supplied to the SA with a copy of the Data Server's public key.

9.3.2.3 Store Self Signed Key

Any SSK produced by the Data Server must be stored securely in a ***Certificate Register*** and cross referenced with the associated Public/Private Key pair.

9.3.3 Validate Certificates

9.3.3.1 Authenticate X509 SA Digital Certificate

This procedure is performed by:

a) Data Servers as part of verifying the SA public key required to authenticate the Data Server Certificate
b) Data Clients to verify the SA public key to be used to authenticate the digital signatures supplied with the ENC data

The Data Server procedure is as follows:

Manually compare the SA public key contained within the SA Digital Certificate with a copy of the printable public key available from the IHO website (http://www.iho.shom.fr). If the above check fails, the Data Server shall not accept the SA Digital Certificate. Otherwise, the SA Digital Certificate is valid and the SA public key it contains can be used in the production of ENC signature files.

9.3.3.2 Authenticate SA signed Data Server Certificate

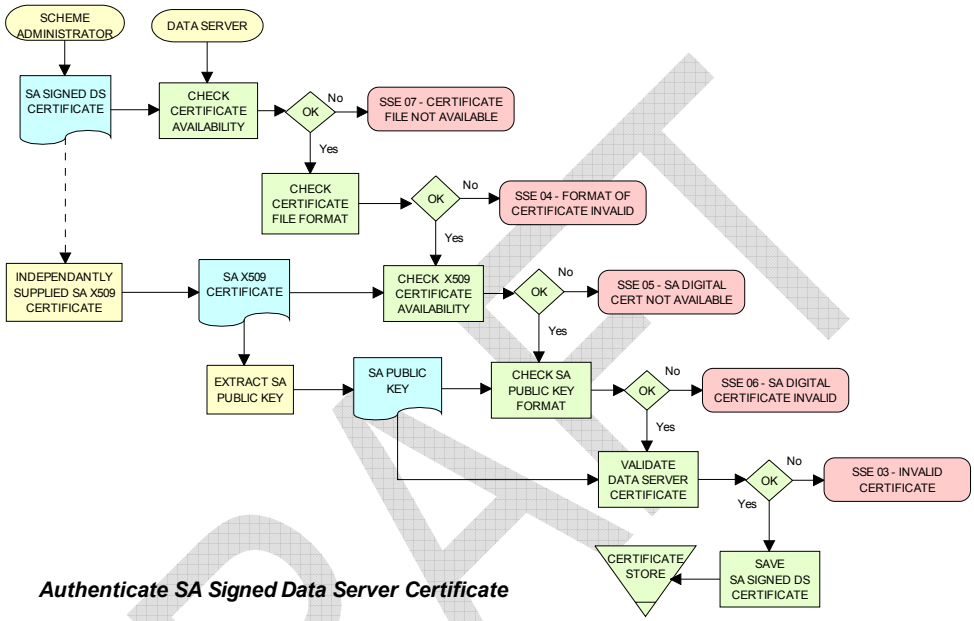
This procedure is performed by Data Servers to authenticate the certificate obtained from the SA before it is used. If Data Servers use an automated means of authentication then the software employed should first check the following:

a) There is a certificate available to authenticate
b) If available, is it in the correct format as per section 5.4.2.6

If a failure is reported in either of these two options the process is to be terminated and an appropriate warning given. Otherwise the process to authenticate should proceed as follows:

a) Obtain the SA public key from the IHO website http://www.iho.shom.fr.

b) Extract the signature elements (i.e. the first two data strings and their attendant headers) from the certificate file. This leaves a public key file.
c) Hash the public key file (obtained from 'b') using the algorithm SHA-1 [3]. All bytes within the file are to be hashed.
d) Verify the signature elements (as removed at 'a' above) by passing it, together with the SA Public Key (the key as obtained in 'a') and the hash of the public key file (as obtained at 'b' above) to the DSA [2]. This will return a status (correct or incorrect).
e) If the Data Server Certificate authenticates correctly, its signature elements 'R' and 'S' may then be used in the construction of ENC digital signatures.



Authenticate SA Signed Data Server Certificate

9.3.3.3 Store SA Signed Data Server Certificate

All Certificates provided by the Scheme Administrator must be stored securely in a ***Certificate Register*** and cross referenced with the associated Public/Private Key pair and SSK.

9.4 Data Management Processes

The Data Management processes includes the creation and management of files for inclusion in an encrypted S-63 exchange set, this includes the following:

a) The PRODUCTS.TXT file (see section 6.2)
b) The SERIAL.ENC file (see section 6.3)
c) The CATD-COMT field of the CATALOG.031 file (see section 6.4.1)
d) Text and Picture file records in the CATALOG.031 file (see section 6.4.2)

Each requires careful management within the Data Server's production software and should be generated in accordance with the formats and conventions described in section 6.

9.5 Encryption, Compression and ENC Signing Processes

9.5.1 Management of Encryption Cell Keys (ECK)

Each ENC is encrypted using a unique cell key and each ENC permit has the capability to store two encrypted cell keys. These keys may be incremented from time to time at the discretion of the Data Server therefore it is important to manage them in an efficient and effective manner.

To create new cell keys and increment existing ones the Data Server will require an application to automatically manage the keys and store them securely. This application must have a method of generating random strings of the correct length and ideally a means of checking that duplicate cell keys are not produced within a set.

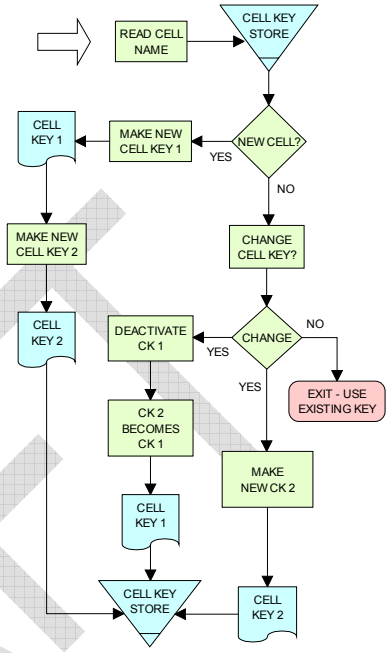
The application must be able to create new cell keys as well as manage the incrementing of those cell keys already in service. The following steps show the logical processes associated with key management, the diagram across is used to further illustrate this.

1. Get cell name and, if necessary, the edition number and determine whether it is a new cell.
2. If new cell make new cell key 1 & 2, if not go to 4?
3. Store new keys in the Key Store.
4. If not a new cell does the key require changing? If no go to 5, if yes go to 6.
5. Exit and keep using the existing cell keys.
6. Cell key 1 is now deactivated and cell key 2 now becomes cell key 1 and is flagged as such in the Key Store.
7. Create new cell key 2 and add to Key Store.

NOTE: The incrementing of the cell keys is at the discretion of the Data Server and is based on the business rules associated with service delivery.

Examples of when keys could be incremented as follows:

- The current encryption keys have been compromised.
- Annually or at an interval defined by the Data Server.
- Synchronized with the issue of a cell new edition.



Data Server Process
Create and Manage Cell Keys

9.5.1.1 Cell Key Format

Unencrypted cell keys are 5 bytes long or 10 hexadecimal characters as shown in the example below:

Cell Key 1	C1CB518E9C	5 bytes
Cell Key 2	421571CC66	5 bytes

9.5.2 Compress ENC file (base or update files)

This procedure is normally performed by the Data Server on ENC files before they are encrypted. The procedure is as follows:

- Compress the ENC cell file using the ZIP standard [6] documented at (www.pkware.com).

The resulting compressed ENC file is used as input to the Encryption stage of the scheme. Only the ENC cell files (base and update) are compressed. This process is always completed before the data is encrypted and signed.

9.5.3 Encrypt ENC Files

9.5.3.1 Base Cell File

This procedure is performed by the Data Server. The ENC file must be compressed before it is encrypted. The procedure is as follows:

a) Select the ***Cell Key*** to be used for encryption (see conditions at 9.5.1).
b) Encrypt the ENC file using the ***Blowfish*** algorithm with the ***Cell Key*** (from 'a') to create an encrypted ENC file.

9.5.3.2 ENC Update File

This procedure is performed by the Data Server. The ENC update file must be compressed before it is encrypted. The procedure is as follows:

a) Select the ***Key*** used to encrypt the ENC base cell file to which the update applies.
b) Encrypt the ENC update file using the ***Blowfish*** algorithm with the ***Key*** (from 'a') to create an encrypted ENC update file.

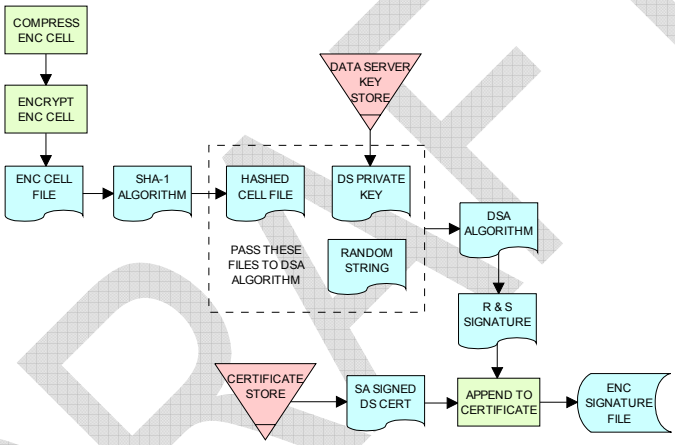
9.5.4 Sign ENC File (Base Cell or Update)

This procedure is performed by Data Servers to digitally sign their ENC data files. The ENC files must be compressed (section 2 & 9.5.2) and encrypted (section 3 & 9.5.3) before they are signed. The procedure is as follows:

a) Hash the encrypted ENC file using the SHA-1 [3] algorithm.
b) Pass Data Server private key and the hashed encrypted ENC file contents to the DSA algorithm [2]. This will return two signature parameters (R and S).
c) Write these as the first two data strings within a signature file compliant with the format and naming convention defined in Section 5.4. The remainder of the file is to be composed of the Data Server Certificate that contains the public key associated with the private key used to create the signature.

Commentaire [p1] : The hashing is part of the algorithm, not a separate step.

Commentaire [p2] : Alter diagram to take out hashing as a separate part of the algorithm.



Process to Create ENC Signature Files

9.5.5 Issue S-63 Encrypted ENC Data

Data Servers will issue S-63 encrypted exchange sets in accordance with the business rules aligned to their data provision services.

9.6 Licensing Processes

9.6.1 Decrypt User Permit

This procedure is performed by the Data Server to extract the HW_ID (unique system identifier) in order to produce cell permits for the Data Client system. The structure of the User Permit is defined in Section 4.2.1. The Procedure for decryption of the User Permit is as follows:

a) Extract M_ID (4 hex characters) from the User Permit.
b) Extract the Check Sum (8 hex characters) from the User Permit.
c) Hash the Encrypted HW_ID (the first 16 characters of the User Permit) using the algorithm CRC32.
d) Compare the outputs of 'b' and 'c'. If they are identical, the User Permit is valid. If the two results differ the User Permit is invalid and the HW_ID cannot be obtained.
e) If the User Permit is valid, convert the Encrypted HW_ID to 8 bytes.

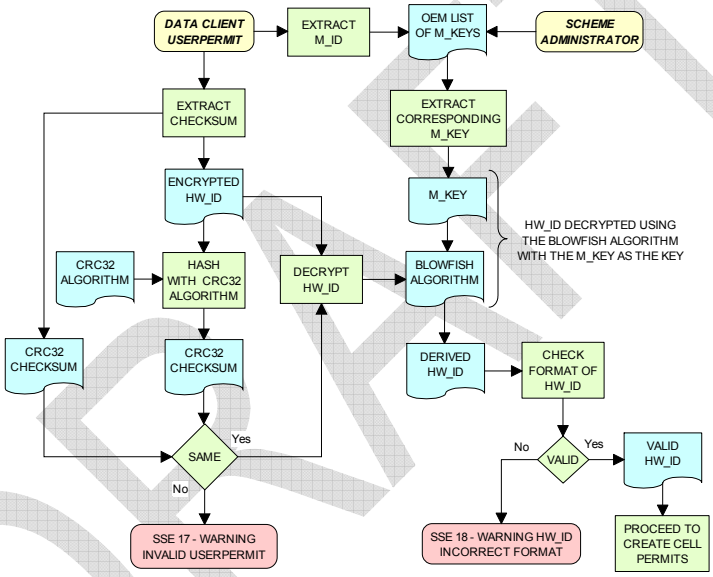
f) Decrypt the Encrypted HW_ID using the Blowfish algorithm with M_KEY as the key. The output will be HW_ID.

Data Servers should confirm that any derived HW_IDs are of the correct length as defined in section 4.2.2.

Example:

User Permit	**73871727080876A07E450C043031**
M_KEY	**3938373635 (ASCII)**

Output from 'a'	**3031**	Extracted M_ID in hexadecimal
Output from 'b'	**7E450C04**	Extracted check sum in hexadecimal
Input to 'c'	**73871727080876A0**	The bytes are given to the hash function left hand byte first (i.e. 73, then 87, then 17 etc)
Output from 'c'	**7E450C04**	Check Sum of Extracted Encrypted HW_ID in hex.
Output from 'f'	**3132333438**	HW_ID in hexadecimal.



Data Server Process - Extract HW_ID from Userpermit

9.6.2 Create Cell Permit

The process to create cell permits is performed by Data Servers based on a Data Client's request. The following process is used to generate Cell Permits in accordance with the structure defined in Section 4.3.

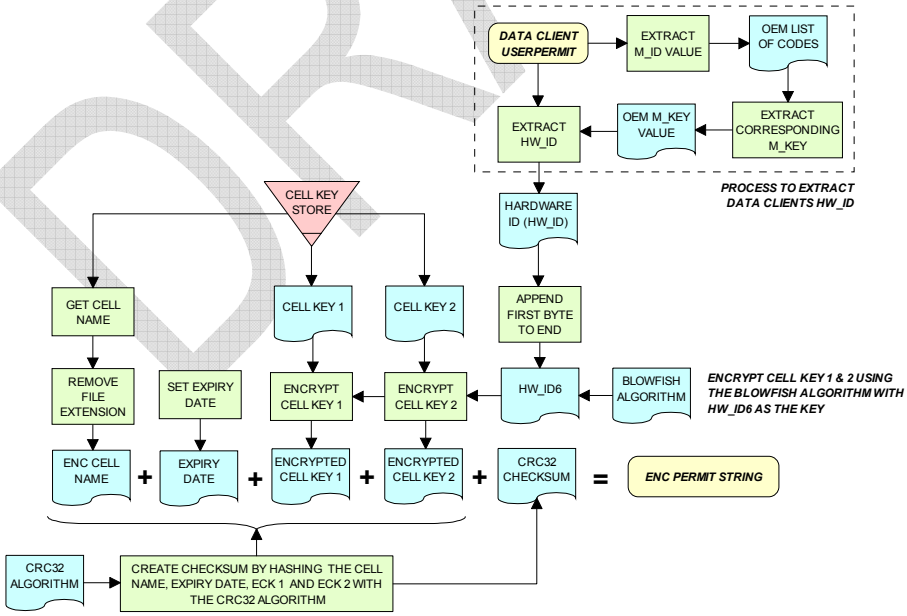
a) Remove the file extension from the name of the ENC file. This leaves 8 characters and is the Cell Name of the Cell Permit.
b) Append the licence Expiry Date, in the format YYYYMMDD, to the Cell Name from 'a'.
c) Append the first byte of HW_ID to the end of HW_ID to form a 6 byte HW_ID (called HW_ID6). This is to create a 48 bit key to encrypt the cell keys.
d) Encrypt Cell Key 1 using the Blowfish algorithm with HW_ID6 from 'c' as the key to create ECK1.
e) Convert ECK1 to 16 hexadecimal characters. Any alphabetic character is to be in upper case.
f) Append to 'b' the output from 'e'.
g) Encrypt Cell Key 2 (CK2) using the algorithm Blowfish with HW_ID6 as the key creating ECK2.
h) Convert ECK2 to 16 hexadecimal characters. Any alphabetic characters are to be in upper case.
i) Append to 'f' the output from 'h'

j) Hash the output from 'i' using the algorithm CRC32. Note the hash is computed after it has been converted to a hex string as opposed to the User Permit where the hash is computed on the raw binary data.
k) Encrypt the hash (output from 'j') using the Blowfish algorithm with HW_ID6 as the key.
l) Convert output from 'k' to a 16 character hexadecimal string. Any alphabetic character is to be in upper case. This forms the ENC Check Sum.
m) Append to 'i' the output from 'l'. This is the Cell Permit.

Example:

HW_ID	`3132333438`	5 bytes in hexadecimal
CK1	`C1CB518E9C`	5 bytes in hexadecimal
CK2	`421571CC66`	5 bytes in hexadecimal
Cell Name	`NO4D0613.000`	Valid S-57 cell name including file extension
Expiry Date	`20000830`	Format YYYYMMDD

Output from 'a'	`NO4D0613`	This is the Cell Name
Output from 'b'	`NO4D061320000830`	Cell name + Expiry date
Output from 'c'	`313233343831`	This is the HW_ID6 in hexadecimal.
Output from 'd' or 'e'	`BEB9BFE3C7C6CE68`	This is ECK1 in hexadecimal
Output from 'f'	`NO4D061320000830BEB9BFE3 C7C6CE68`	Cell name + expiry date + ECK1
Output from 'g' or 'h'	`B16411FD09F96982`	This is ECK2 in hexadecimal
Output from 'i'	`NO4D061320000830BEB9BFE3 C7C6CE68B16411FD09F96982`	Cell name + expiry date + ECK1 + ECK2
Input to 'j'	`NO4D061320000830BEB9BFE3 C7C6CE68B16411FD09F96982`	The ASCII values of the output from 'i' (36 bytes in total). The bytes are given to the hash function left hand byte first (i.e. xx, then xx, then xx etc).
Output from 'j'	`780699093`	CRC32 of 'j' 4 byte number
Output from 'k'	`8 byte non-printable`	Encrypted CRC32
Output from 'l'	`795C77B204F54D48`	Encrypted CRC32 in hexadecimal
Cell Permit	`NO4D061320000830BEB9BFE3C7C6CE68B16411FD09F96982795C77B204F54D48`	



Data Server Process - Create Cell Permit

9.6.3 Issue ENC Licences

Data Servers will issue ENC Licences to access S-63 encrypted ENC in accordance with business rules aligned to their data provision services. Data Servers will make the details of their services available to Data Clients before licences are issued.

9.7 Security QA Procedures – Data Server

9.7.1 Data Protection Scheme Information

The SA will provide copies of all information required to operate the Data Protection Scheme to a Data Server.

9.7.2 System Compliance Testing

The Data Server must perform internal compliance testing of their implementation of the protection scheme, based on the descriptions provided in this document and the supplied test data.

9.7.3 Storage of M_IDs and M_KEYs

When the Data Server joins the scheme, the SA shall provide the Data Server with the proprietary M_ID and M_KEY information for all participating manufacturers. The SA shall immediately inform all Data Servers about any amendments to the list of M_ID and M_KEYs as new manufacturers join the scheme.

The receipt of all M_IDs and M_KEYs by the Data Server are to be recorded securely in an **M_ID / M_KEY Register**.

9.7.4 Acceptance and Checking of the SA Digital Certificate (and Public Key)

A Data Server will receive the SA public key in two formats, as an X.509 Digital Certificate and as a printable public key. The Data Server shall have the capability to load the SA digital certificate and manually compare the public key against the printed public key. The Data Server shall only accept the SA public key when this has been done. This process applies to the original SA public key and to any subsequent keys issued by the SA.

The Data Server shall maintain records, in a **SA Public Key Register**; of what SA public keys have been used. This should contain a copy of each key as well as the date on which it was issued.

9.7.5 Creation of Digital Signature Keys (Private and Public keys)

The Data Server shall have the ability to create its own private and public key pair as detailed in section 9.3.

The private key must be stored securely with access limited to only those persons who have a need to know. The Data Server will create a new public/private key pair and request a new Data Server Certificate from the SA if its private key is compromised.

The Data Server shall create a self signed key (SSK) and send it to the SA for conversion into a Data Server certificate. Upon receipt, the SA will contact the sending Data Server to confirm that the delivered SSK did originate from its stated source.

9.7.6 Acceptance of the Data Server Certificate from the SA

The Data Server shall verify and securely store the Certificate returned by the SA by following the process laid out in section 9.3.3.3.

9.7.7 Creation of Cell Keys

The Data Server shall have the ability to create and manage Cell Keys as defined in the section 9.5.1. The Data Server is responsible for ensuring that cell keys are securely stored once created.

9.7.8 Compression, Encryption and Signing S-57 data

The Data Server shall have the ability to compress, encrypt and sign ENC information as defined in sections 9.5.2, 9.5.3, and 9.5.4. Access to the signing program should be restricted to only those authorised to release data.

9.7.9 Creation of Random Values

In order to sign ENC information, the Data Server will create random values. The Data Server shall ensure that the same value is not be used for any two separate signatures.

9.7.10 Creation of Cell Permits

The Data Server must have the ability to create a Cell Permit for a Data Client. The Data Server must issue a new Cell Permit to its Data Clients when an ENC cell is encrypted with a different cell key (e.g. when it is issued as a new edition).

9.7.11 Decryption of User Permits

The Data Server must have the ability to decrypt User Permits to obtain the Data Client HW_ID. The HW_ID is required by the Data Server to create a Cell Permit.

10 OEM and DATA CLIENT PROCESSES

10.1 Data Clients

Data Clients are the users of ENC information and will receive protected information from Data Servers. The OEM is responsible for developing software applications capable of authenticating the ENC digital signatures and decrypting the ENC information in compliance with the procedures defined in the protection scheme. Navigators with ECDIS/ECS systems are examples of Data Clients.

10.2 Original Equipment Manufacturers (OEMs)

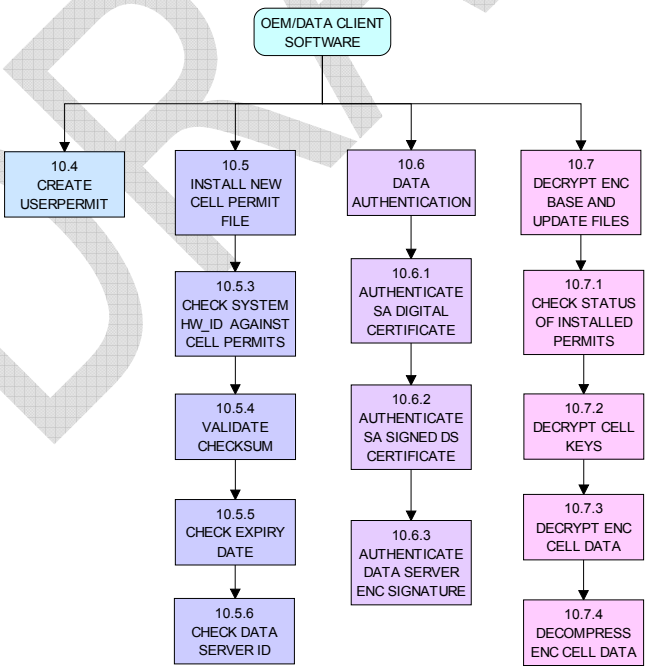
Manufacturers subscribing to the data protection scheme must build software routines in support of the scheme. The S-63 standard contains specifications and test data for validating the manufacturer's software application. The SA will provide the manufacturer with a unique set of manufacturer codes (M_KEY & M_ID).

The manufacturer must also provide a secure mechanism within their software systems for uniquely identifying each end user installation. The data protection scheme requires each installation to have a unique hardware identifier (HW_ID). The Data Servers will use the M_KEY and HW_ID information to issue encrypted ENC cell keys to a Data Client specific installation. Each ENC is encrypted with a unique cell key however; by encrypting these using the Data Client's unique HW_ID this ensures that they cannot be transferred between several ECDIS from the same manufacturer.

The manufacturer is required to cooperate in the protection of the ENC information within end user systems

10.3 OEM & Data Client Processes

The main responsibilities of approved OEMs and their software applications are depicted in the following diagram. Each *"Process Box"* cross references the particular section where these operations are detailed.



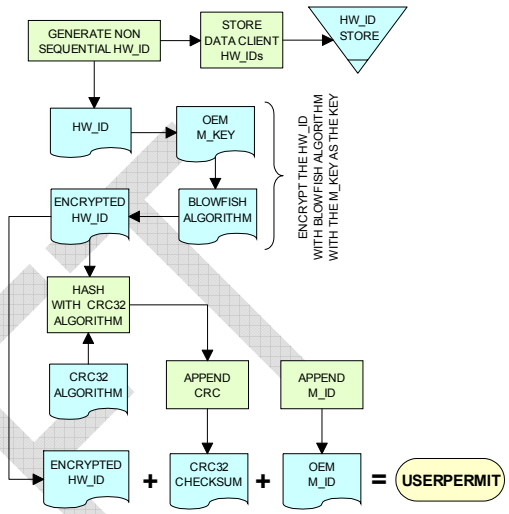
Main OEM/Data Client Processes

10.4 Create Data Client Userpermit

This procedure is performed by the system manufacturer (OEM) who creates a Data Client specific userpermit. This is provided to the Data Client usually when purchasing an ECS/ECDIS. This userpermit enables Data Clients to obtain cell permits from Data Servers. The cell permits are generated using the encrypted HW_ID that is contained within the userpermit. The format and structure of the User Permit is defined in section 4.2.

The procedure for creating a User Permit is as follows:

a) Encrypt HW_ID using the Blowfish algorithm with M_KEY as the key.
b) Convert the resultant value to a 16 character hexadecimal string. Any alphabetic character should be in upper case.
c) Hash the 16 hexadecimal characters using the algorithm CRC32.
d) Convert output from 'c' to an 8 character hexadecimal string. Any alphabetic characters should be in upper case. This is the Check Sum.
e) Append to 'b' the output from 'd'.
f) Convert the M_ID to a 4 character string. Any alphabetic characters should be in upper case.
g) Append to 'e' the output from 'f'. This is the User Permit.



OEM - Create Userpermit

Example:

HW_ID	**3132333438 (ASCII)**
M_KEY	**3938373635 (ASCII)**
M_ID	**3031 (ASCII)**

Expected Results:

Input to 'a'	**3132333438** and **3938373635**	HW_ID and M_KEY in hexadecimal.
Output from 'a'	8 bytes	Non-printable.
Input to 'c'	**73871727080876A0**	The hexadecimal values of the above string. The bytes are given to the hash function left hand byte first (i.e. 73, then 87, then 17 etc)
Output from 'c'	**7E450C04**	CRC32 result in hexadecimal
Output from 'e'	**73871727080876A07E450C04**	CRC32 result appended to the encrypted HW_ID
Output from 'f'	**3031**	Result appended to encrypted HW_ID & CRC32
User Permit	**73871727080876A07E450C043031**	

10.5 ENC Cell Permit Installation

New Cell permits are delivered to a Data Client's system in a file named PERMIT.TXT. The structure and format of this file are given in section 4.3. The Data Clients system must be able to read in this file and perform a number of checks. Each cell permit record contains a Data Server ID that enables OEMs to manage permits and data in a multi supplier environment. The following sections outline how this file must be managed and the checks that must be carried out when installing a new permit file.

10.5.1 Check for a Cell Permit File

The Data Client system must first check that a valid cell permit file is available to install. A facility should be available for Data Clients to browse to a specific location on the system where the PERMIT.TXT file is available to install. If any text file other than one named PERMIT.TXT file is selected the system should return a warning as follows:

"SSE 11 - Cell permit file not found"

10.5.2 Check Cell Permit Format

If a valid PERMIT.TXT is located the system must then check that the format of the file is correct as defined in section 4.3. If not the data client must inform the user as follows:

"SSE 12 - Cell permit format is incorrect".

10.5.3 Check the HW_ID

Data Client system must check that the HW_ID encoded into the dongle/software security device is comparable with the HW_ID encrypted in the cell permits. If the values are the same the system shall continue with the checks below, if not an error message must be returned as follows:

"SSE 19 - Permits are not valid for this system. Contact your data supplier to obtain the correct permits".

10.5.4 Check Cell Permit Check Sum

This procedure is performed by the Data Client system and is comprised of the following steps.

a) Extract the last 16 hex characters (ENC Check Sum) from the Cell Permit.
b) Convert these 16 hex characters to 8 bytes.
c) Hash the remainder of the Cell Permit as left after 'a' using the algorithm CRC32.
d) Append the first byte of HW_ID to the end of HW_ID to form a 6 byte HW_ID (called HW_ID6).
e) Encrypt the hash (output from 'c') using the Blowfish algorithm with HW_ID6 as the key.
f) Compare the output from 'e' with the output from 'b'. If they are the same, the Cell Permit is valid. If they differ, the Cell Permit is corrupt and Cell Permit is not to be used.

Example:

HW_ID	`3132333438`	in hexadecimal
Cell Permit	`NO4D061320000830BEB9BFE3C7C6CE68` `B16411FD09F96982795C77B204F54D48`	Example cell permit

Output from 'a'	`795C77B204F54D48`	In hexadecimal
Output from 'b'	8 byte non-printable	Encrypted CRC32
Input to 'c'	`NO4D061320000830BEB9BFE3` `C7C6CE68B16411FD09F96982`	Cell permit after removal of 16 hex encrypted CRC32 The bytes are given to the hash function left hand byte first (i.e. xx, then xx, then xx etc).
Output from 'c'	`780699093`	4 byte CRC32 of cell permit after removal of 16 hex encrypted CRC32
Output from 'd'	`313233343831`	This is HW_ID6
Output from 'e'	8 byte non-printable	Encrypted CRC32

If the calculated CRC32 value is not the same as the value contained in the cell permit the system must inform the Data Client as follows:

"SSE 13 Cell Permit is invalid (checksum is incorrect)"

The system must not install any invalid permits.

10.5.5 Check Cell Permit Expiry Date

When installing a new PERMIT.TXT file the Data Client system must check that the permits being installed have not expired. The system must check that the expiry date of each permit against the system date (Computer Clock) and if available the time from the GPS receiver/signal. If the permits have expired the following message should be displayed as follows:

"SSE 15 - Subscription service has expired. Please contact your data supplier to renew the subscription licence."

NOTE: The system may install expired/valid permits but any cells subsequently displayed in the viewer under these conditions **MUST** display a permanent warning to the user as follows:

"SSE 25 - The ENC Permit for this cell has expired. This cell may be out of date and MUST NOT be used for NAVIGATION."

See section 10.7.1.1 for checking the expiry date at load time.

If the expiry date of the permit is in advance of the computer clock/GPS signal then a further check must be made to see how long the licenced subscription has to run. If this is 30 days or less then the system should give a warning informing the Data Client as follows:

"SSE 20 - Subscription service will expire in less than 30 days. Please contact your data supplier to renew the subscription licence."

The Data Client can then take steps to renew the licence before it expires. The system should then proceed to install the permits. If the permit has more than 30 days before expiring the permits should be installed without warning.

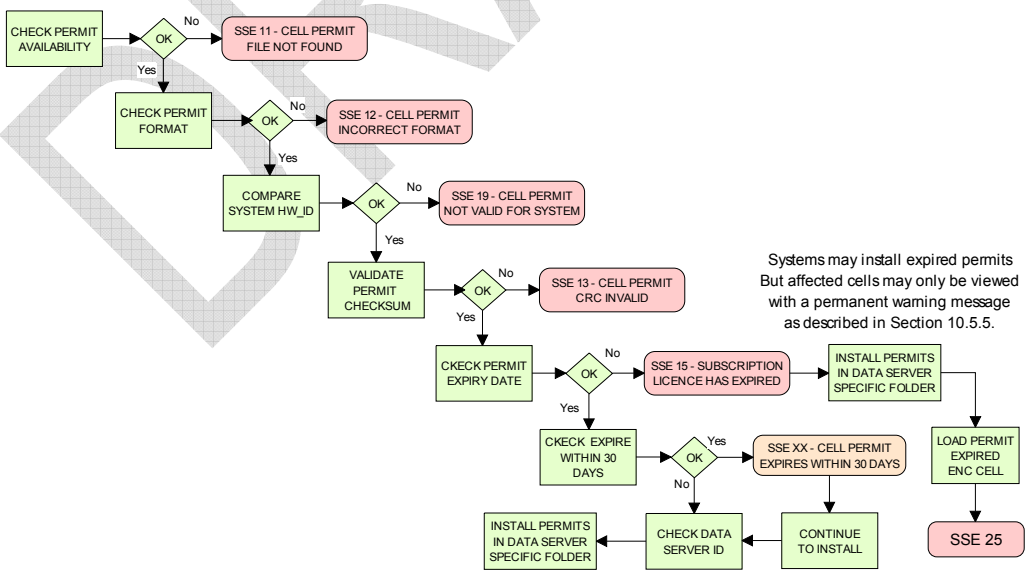
10.5.6 Check Data Server ID

The S-63 Data Protection Scheme makes takes account of a multiple supplier environment, that is to say Data Clients may obtain licences from more than one Data Server. There are several instances where Data Clients may have ENC data from multiple suppliers as follows:

- Duplicate cells licenced from different Data Servers
- Change from one Data Server to another

It is important that Data Client systems are able to manage these instances. Each permit record contains a Data Server ID field (see section 4.3.3). This field, if filled, contains a two character alphanumeric ID unique to each Data Server assigned by the SA. Since cell permits issued by one Data Server will not necessarily decrypt ENCs supplied by another it is important to maintain an association between the cell permits and encrypted ENCs. OEMs should ensure that their systems are capable of maintaining these associations, e.g. by creating Data Server specific folders where permits are stored.

The Data Server ID for encrypted ENC exchange sets is contained in the SERIAL.ENC file (see section 6.3.1) and is identical to that contained in the cell permit record.



OEM System - Install & Validate Cell Permit

10.6 ENC Authentication and Integrity Checks

OEM systems must be capable of authenticating the source of the encrypted ENC data and validate its integrity. This is achieved in two ways as follows:

- By Authenticating the SA signature held as part of the Data Server Certificate that forms part of the ENC signature file.
- By validating the Data Server ENC signature (corresponding to the ENC Cell Data) in the ENC signature file.

OEMs and Data Clients must first of all confirm that the SA certificate (whether X509 or ASCII format) installed on the ECS/ECDIS is correct and current. This is dealt with in section 10.6.1 below.

10.6.1 Authenticate/Verify SA Digital Certificate

This procedure is performed by OEMs or Data Clients to verify that the SA public key installed on the ECS/ECDIS is correct and current in respect of the IHO S-63 Data Protection Scheme. It is this SA public key that is used to authenticate the SA signed Data Server Certificate supplied by Data Servers as part of the ENC signature file. The procedure is as follows:

Manually compare the SA public key contained within the independently installed SA Digital Certificate with a copy of the printable public key available from the IHO website (http://www.iho.shom.fr). If the above check fails, the system shall not accept the SA Digital Certificate. Otherwise, the SA Digital Certificate is valid and the Data Server public key it contains can be used to authenticate SA signed Data Server Certificate held as part of the ENC signature file.

NOTE: The Data Client must have means by which users can access the installed certificate from the application.

10.6.1.1 Manual Checking of the SA Public Key

The SA public key can be accessed from the IHO website as follows:

www.iho.shom.fr → Home → Publications → Download List → S-63 → S-63 SA Certificate

The following webpage will be displayed:

S-63 DIGITAL CERTIFICATES

Digital Certificates are files that bind a specific public key together with other information to an individual or organisation. The S-63 standard uses a 2-level chain of certificates to operate the data protection scheme.

*The IHB operates as the Scheme Administrator and has issued the root Digital Certificate for use within the protection scheme. The SA certificate used by IHB will be a self-signed certificate. It is available both as a X-509 compliant file **IHO.CRT** and as a text file **Scheme Administrator Public Key.txt**. Both files are contained in an **SA Certificate** compressed file.*

The SA will issue Data Server Certificates to all Data Servers participating in the protection scheme. The Data Server Certificate contains the Data Server Public Key and the SA signature of this Key. Since only the SA can issue Data Server Certificates, the chain of trust can be established by authenticating the SA signature on the Data Server Public Key.

The protection scheme requires the SA public key to be installed on end user systems by all users of the protection scheme. The Data Server Certificate is contained within each signature file and the Data Server Public Key can be trusted if the SA certificate is valid. The installation of the SA certificate (and the public key held within) should be carried out as a separate, independent operation and be subject to carefully controlled operating procedures.

In the second paragraph above click on ***"SA Certificate"*** and a ***"File Download"*** dialog will be displayed which gives the option to ***"Open"*** or ***"Save"*** the zipped file named ***"S-63_SA_Certificate.zip"***. This file contains two files as follows:

1. **IHO.CRT (The X509 Certificate)**

   Opening this file reveals a ***"Certificate"*** dialog, selecting the ***"Details"*** tab and highlighting ***"Public Key"*** displays the IHO public key. The example below is the IHO public at the time this document was published. Note that the first 6 characters [024100] represent the certificate parameters and can be either positive [0240] or negative [024100].

```
0241 0096 3F14 E32B A537 2928 F24F 15B0 730C
49D3 1B28 E5C7 6410 0256 4DB9 5995 B15C F880
0ED5 4E35 4867 B82B B959 7B15 8269 E079 F0C4
F492 6B17 761C C89E B77C 9B7E F8
```

   This character string (minus the certificate parameters) should be compared with the installed certificate to confirm that they are the same. If it is, then the certificate is authenticate, if not, it should be rejected.

2. **Scheme Administrator Public Key.txt**

   Opening this file displays the following SA public key parameters.

```
// BIG p
FCA6 82CE 8E12 CABA 26EF CCF7 110E 526D B078 B05E DECB CD1E B4A2 08F3 AE16 17AE
01F3 5B91 A47E 6DF6 3413 C5E1 2ED0 899B CD13 2ACD 50D9 9151 BDC4 3EE7 3759 2E17.
// BIG q
962E DDCC 369C BA8E BB26 0EE6 B6A1 26D9 346E 38C5.
// BIG g
6784 71B2 7A9C F44E E91A 49C5 147D B1A9 AAF2 44F0 5A43 4D64 8693 1D2D 1427 1B9E
3503 0B71 FD73 DA17 9069 B32E 2935 630E 1C20 6235 4D0D A20A 6C41 6E50 BE79 4CA4.
// BIG y
963F 14E3 2BA5 3729 28F2 4F15 B073 0C49 D31B 28E5 C764 1002 564D B959 95B1 5CF8
800E D54E 3548 67B8 2BB9 597B 1582 69E0 79F0 C4F4 926B 1776 1CC8 9EB7 7C9B 7EF8.
```

   If this file is used for authentication it should be checked against the installed certificate or public key file. If checking against an installed certificate then only the ***"BIG y"*** string should be verified to see if it is the same. If checking against SA public key file then all parameters must be verified to see if it is the same. In either case if the file is correct then the public key is authenticated, if not, it must be rejected.

10.6.2 Authenticate SA signed Data Server Certificate

This procedure is performed by the Data Client's system to authenticate the SA signed Data Server Certificate stored as part of the ENC signature file against the installed SA public key. This process is carried out before the Data Server public key is extracted to authenticate the ENC signature. Refer to section 5.3.2 for the structure of signature/certificate pairs in a signature file.

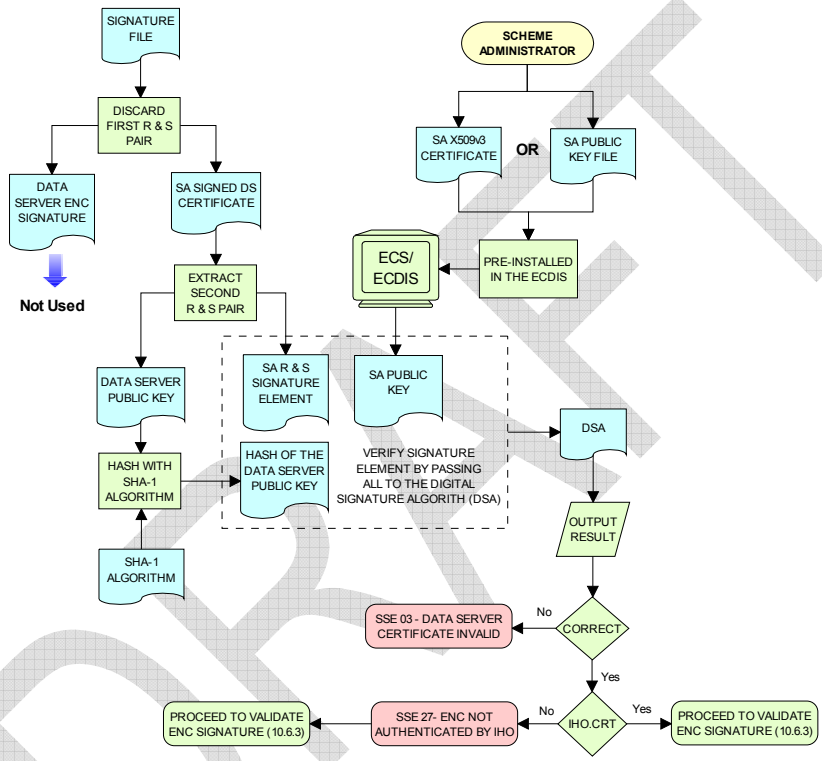
Prior to the authentication process the system must first check the availability, format and status of the certificate or public key installed on the system. If there are any problems this should be reported to the data client in a meaningful way as follows:

1. The SA certificate or public key is not present on the system (SSE 05 and terminate process).
2. The format of the SA certificate or public key is incorrect (SSE 08 and terminate process).
3. The SA certificate has expired (SSE 22 and terminate process).

The authentication procedure is outlined below:

a) Extract the ENC signature file.

b) Discard the first signature part (i.e. the first two data strings and their attendant headers. This is the Data Server signature of the ENC data). This leaves the SA signed Data Server Certificate.
c) Extract the remaining signature part (i.e. the first two data strings and their attendant headers from the remaining file obtained from 'b'). This leaves a public key file.
d) Hash the public key file (obtained from 'c') using the algorithm ***SHA-1*** [3]. All bytes within the file are to be hashed.
e) Verify the signature part (as removed at 'c' above) by passing it (the signature), together with the SA Public Key file (the key) and the hash of the public key file (obtained at 'd') to the ***DSA*** [2]. This will return a status (correct or incorrect).



Authenticate SA Signed Data Server Certificate

10.6.2.1 Authentication against non-SA signed Data Server Certificate

There may be instances where there is more than one certificate or public key stored on the data client. This may be especially so during the transition to the correct use of the S-63 scheme. Therefore a check is necessary to ensure that the data server certificate authenticates correctly with the IHO.CRT or IHO.PUB installed on the data client.

If the data server certificate authenticates against anything other than the IHO.CRT or IHO.PUB stored on the data client then a warning message **MUST** be displayed as follows:

"SSE 27 - "This ENC is not authenticated by the IHO acting as the Scheme Administrator"

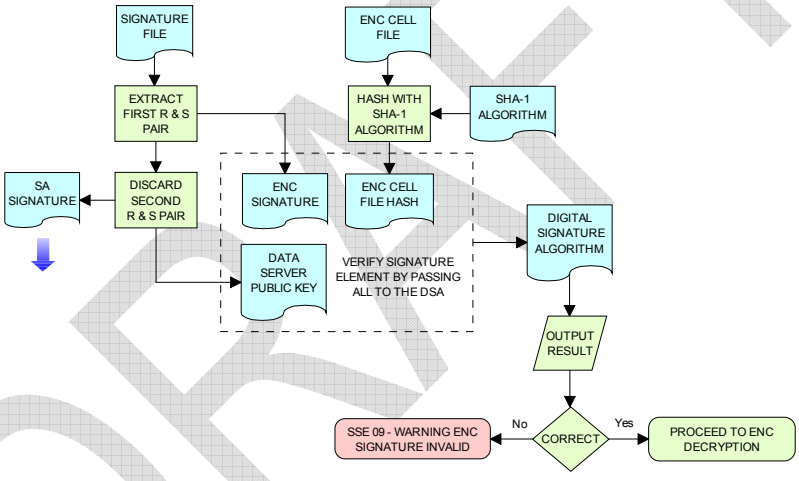
If this message is displayed the data client should still continue to the next stage of authentication ENC signature authentication) and decryption.

10.6.3 Authenticate ENC Cell File

This procedure is performed by Data Client's systems to validate the ENC signature (held in the ENC signature file) corresponding to a specific ENC cell file. It is expected that the Data Client has already performed the procedures to authenticate the SA digital certificate (section 10.6.1) and the Data Server Certificate within the signature file (section 10.6.2). The procedure to authenticate the ENC Cell File is as follows:

a) Extract the ENC signature file uniquely related to an ENC cell file.
b) Extract the first signature part (i.e. the first two data strings and their attendant headers). This leaves the certificate.
c) Discard the remaining signature part (i.e. the first two data strings and their attendant headers from the remaining file). This leaves a public key file.
d) Hash the associated ENC Cell File using the algorithm ***SHA-1*** [3]. All bytes within the file are to be hashed.
e) Verify the signature part (as extracted at 'b' above) by passing it (the signature), the public key - as left at 'c' above (the key) and the hash of the ENC Cell File, as obtained at 'd' above, to the ***DSA*** [2]. This will return a status (correct or incorrect).

If the ENC signature is not authenticated correctly, the Data Client shall not decrypt the ENC because its origins cannot be verified. If the ENC is authenticated correctly, the ENC can safely be decrypted.



Authenticate ENC Cell File - Validate ENC Signature

10.7 Decrypt ENC Base Cell and Update Files

Before decrypting new ENC base cells and update files the system should first check the subscription status of installed cell permits. This process is to determine whether the Data Client is licenced to receive and install new ENC data. It also seeks to give the Data Client adequate warning messages prior to the expiry of the licence.

10.7.1 Check Subscription Status of Installed Permits

Section 10.5 identified the processes and checks that are carried by the Data Client's system when installing cell permits. This section determines how cell permits are managed by a Data Client's system once installed. It is also designed to give Data Clients advanced warning of subscription permits that are about to expire, especially when ENC data is being used for navigation.

10.7.1.1 Check if Subscription has expired in a Cell Permit – Required Warning

This check is performed on new ENC base cells and update files prior to decryption. This check is required to inform the Data Client that the subscription licence has expired but that additional ENC

updates/base cells have become available. The warning is only applicable for subscription licenses and is not to be used for single purchase licenses, ref. section 4.3.3. The procedure is outlined in the flowchart below and the subsequent step by step description:

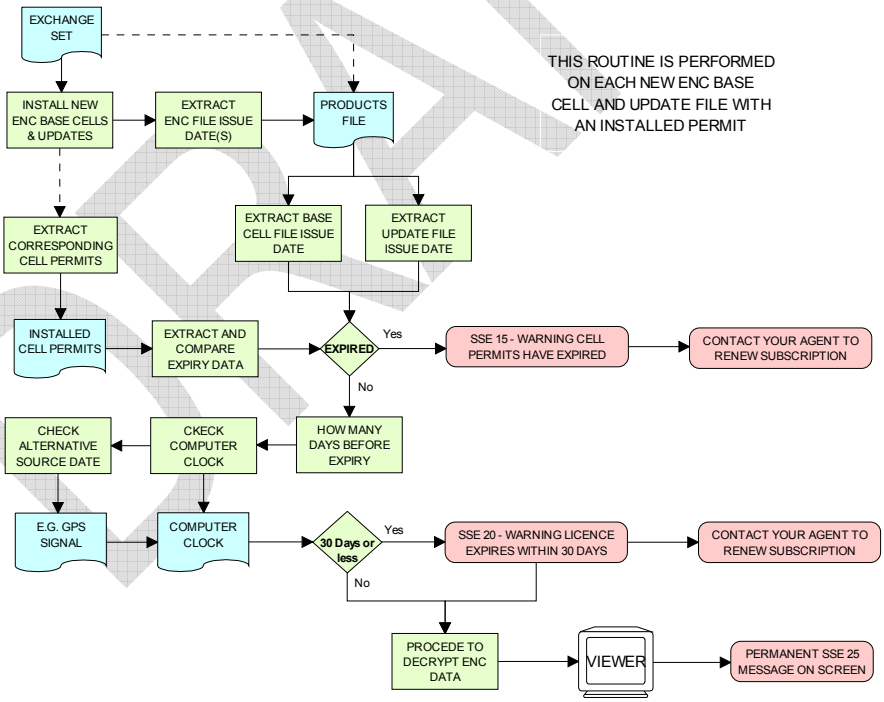
a) Extract expiry date of the loaded ENC Cell Permit corresponding to the ENC file to be decrypted.
b) Extract the issue dates of the ENC base cell and latest update file (if available[10]) to be decrypted from the PRODUCTS.TXT file. These are located in the second (Product Issue Date) and fourth (Issue Date of Latest Update) fields of the cell record corresponding to the cell being decrypted.
c) If two dates (in fields two and four) are returned at b) then only the latest date[11] should be used when checking against the expiry date.
d) If the Issue Date of the base cell or the update obtained at b) and c) is newer (in advance of) the permit expiry date obtained at a) the permits are deemed to have expired. A warning message must be displayed as follows:

"SSE 15 - Subscription service has expired. Please contact your data supplier to renew the subscription licence."

The application may install expired ENC permits but must display the ***""SSE 15"*** warning above.

The application **MAY DECRYPT** any ENC data files with an expired cell permit but a permanent warning message **MUST** be displayed informing the user that they may be viewing ENCs that are not up to date. A **PERMANENT** warning message **MUST** be displayed in the viewer as follows:

"SSE 25 - The ENC Permit for this cell has expired. This cell may be out of date and MUST NOT be used for NAVIGATION."



Process to Check Subscription Status before Decryption

[10] If no updates have been issued for a cell there will be no information available.
[11] The "Issue Date of Latest Update" field, if filled, will not always be in advance of the "Product Issue Date", for instance in the case of re-issues.

10.7.1.2 Check Subscription Status – Required 30 day warning

This check must be performed every time new ENC base cell or update files are installed and is required to inform the Data Client on the status of the subscription licence ahead of expiry. The intention is to ensure that the Data Client has time to renew their subscription and obtain an updated Cell Permit from the Data Server. The warning is only applicable for subscription licenses and is not to be used for single purchase licenses, ref. section 4.3.3. The procedure is as follows:

a) Obtain the system date and, if available, any alternative reliable time sources, e.g. GPS signal.
b) Obtain the subscription expiration date from the Cell Permit file.
c) Compare the system date from 'a' and the subscription expiration date from 'b'.
d) If it is 30 days or more before the subscription expires, the system can operate without any further notices to the user.
e) If it is less than *30 days* before the subscription expires, the system may be able to decrypt and uncompress new information issued during the subscription period. The system should issue a warning message to the user e.g.

"SSE 20 - Subscription service will expire in less than 30 days. Please contact your data supplier to renew the subscription licence."

10.7.2 Decrypt the Cell Keys in a Cell Permit

This procedure is performed by the Data Client system after the successful authentication of the ENC signature file. The decrypt process begins with the extraction of the cell keys required to decrypt the ENC and comprises of the following:

a) Append the first byte of the Data Client HW_ID to the end of HW_ID to form a 6 byte HW_ID (called HW_ID6).
b) Extract ECK1 from the Cell Permit and convert this from the 16 character hexadecimal string to 8 bytes.
c) Decrypt the converted ECK1 (output from 'b') using the Blowfish algorithm with HW_ID6 as the key. This will yield CK1.
d) Extract ECK2 from the Cell Permit and convert this from the 16 character hexadecimal string to 8 bytes.
e) Decrypt the converted ECK2 (output from 'd') using the Blowfish algorithm with HW_ID6 as the key. This will yield CK2.

Example:

HW_ID	`3132333438`	In hexadecimal
Cell Permit	`NO4D061320000830BEB9BFE3C7C6CE68` `B16411FD09F96982795C77B204F54D48`	Example of cell permit

Output from 'a'	`313233343831`	HW_ID6
Output from 'b'	8 byte non-printable	Encrypted ECK1
Output from 'c'	`C1CB518E9C`	Cell key 1 (hex)
Output from 'd'	8 byte non-printable	Encrypted ECK2
Output from 'e'	`421571CC66`	Cell key 2 (hex)

Note that the unencrypted Cell Keys are 5 bytes in length even though the encrypted cell keys are 8 bytes in length. This is because blowfish pads the Cell Keys to 8 bytes in length when it encrypts them and it un-pads the Encrypted Cell Keys when it decrypts them.

10.7.3 Decrypt ENC Base Cell or Update File

This procedure is performed by the Data Client's system and is carried out as outlined in the flowchart (for sections 10.7.2 and 10.7.3) and the step by step guide below[12]:

a) Decrypt the ENC file using the Blowfish algorithm with CK1 as the decryption key[13].

[12] OEMs should note that there is no requirement to check the edition date against the permit or words to this effect.
[13] Rather than decrypting and decompressing the entire ENC file the data client can check that the decrypted header information is compliant with the ZIP standard [6].

b) Decompress the ENC file. If decompression is successful, the ENC file is decrypted and ready for import.
c) If decompression is unsuccessful, decrypt the ENC file using the Blowfish algorithm with CK2 as the decryption key.
d) Decompress the ENC file. If decompression is successful, the ENC file is decrypted and ready for use.
e) If decompression is unsuccessful in 'b' and 'd', this means that the Cell Permit does not contain any valid cell keys. The system should return a relevant warning message and advise the Data Client that a new Cell Permit should be obtained from the Data Server.

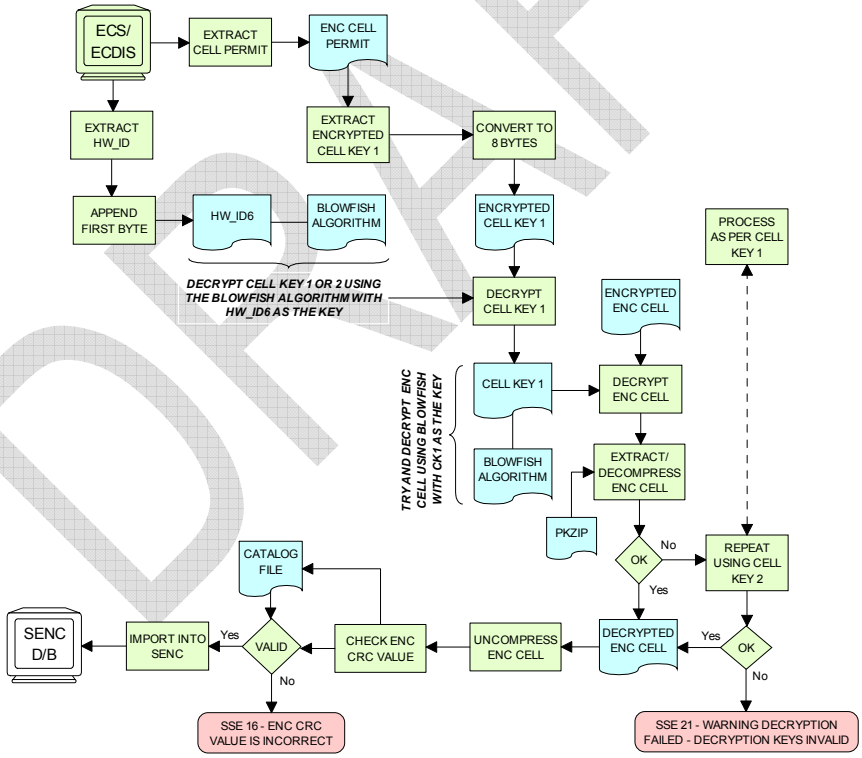
"SSE 21 – Decryption failed no valid cell permit found. Permits may be for another system or new permits may be required, please contact your supplier to obtain a new licence."

10.7.4 Decompress ENC file (base cell or update)

This procedure is performed by the Data Client on decrypted ENC files. The procedure is as follows:

Uncompress the ENC file using the ZIP standard [6] to create a file fully compliant with the S-57 Edition 3.1 ENC Product Specification.

NOTE: The CRC value of the ENC [1] is always computed on the unencrypted ENC information. The application must confirm successful decryption and decompression by conducting the CRC check on all ENC information.



Decrypt & Uncompress ENC Base Cell and Update Files

10.8 QA Procedures – Data Client

10.8.1 Acceptance and Checking of the SA Digital Certificate (and Public Key)

A Data Client will receive the SA public key in two formats, as an X.509 Digital Certificate and as a printable public key. The Data Client shall have the capability to load the SA digital certificate and manually compare the public key against the printed public key (see section 10.6.1.1). The Data Client shall only accept the SA public key when this has been done. This process applies to the original SA public key and to any subsequent public keys issued by the SA.

10.8.2 Creation of User Permit

The system/application suppliers shall be able to create their own User Permit containing the encrypted HW_ID. The User Permit will be provided to Data Servers who will then create Cell Permits for the requested ENC information. A User Permit shall only be created to request Cell Permits from a Data Server.

10.8.3 Verification of Data Server Certificate

The Manufacturer application shall allow the verification of a Data Server Certificate contained within an ENC signature file using the SA public key. If the Data Server Certificate is verified successfully, the application shall then extract the Data Server public key from the Data Server Certificate and use it to verify the ENC signature.

The SA will inform the Manufacturer about revoked Data Server Certificates.

10.8.4 Validation of Cell Permits

The Data Client system must have the ability to validate the integrity of a Cell Permit by checking the encrypted check sum. This shall be done by following the procedure set out in section 10.5.4 of the specification.

The Data Client must be able to manage Cell Permits provided by several Data Servers. The Data Client must also be able to manage Cell Permits for the same ENC provided by multiple Data Servers.

The Data Client must have the ability to manage stored Cell Permits so that old ones can be deleted and new ones added to, or merged with, those stored.

The Data Client application should not allow the Data Client to be able to view or copy the decrypted cell keys.

10.8.5 Authentication and Decryption of ENC Information

The Data Client must be able to accept a signed and encrypted ENC data set by following the procedure defined in sections 10.6 and 10.7.

10.9 *QA Procedures – Manufacturers (OEMs)*

10.9.1 Confidentiality Agreement

The SA will provide a manufacturer with copies of all information required to operate the Data Protection Scheme within a Confidentiality Agreement. The Manufacturer shall abide by the terms and conditions of the Confidentiality Agreement and ensure that all supplied information is kept up to date.

10.9.2 System Compliance Testing

The Manufacturer shall perform internal compliance testing of their implementation of the protection scheme, based on the descriptions provided in this document and the supplied test data.

The SA will only issue M_IDs and M_KEYs on successful compliance as provided by a self certification document.

10.9.3 Storage of M_IDs and M_KEYs

When the Manufacturer has joined the scheme, the SA shall provide the proprietary M_ID and M_KEY information for the creation of User Permits.

The users of the Manufacturer application must not be able to view or extract the M_KEY information.

10.9.4 Creation of HW_IDs

The Manufacturer shall have the ability to create HW_IDs of the format required within the standard. These are to be random so that they will not be sequential and cannot be duplicated.

The users of the Manufacturer application must not be able to view or extract the HW_ID information from the application.

10.9.5 Recording of HW_IDs

The Manufacturer must record, in an **HW_ID Register**, the values of each HW_ID created. These details are to be made available to the SA upon request.

Page intentionally left blank

11 S-63 Error Codes and Explanations

The following error codes and messages are defined in the flowcharts in sections 8, 9, and 10. It is expected that application developers support the error conditions with an appropriate error message. When an error occurs, this can in some instances, prevent further processing of the data.

Error Code	Error/Warning Message
SSE 01	*"Self Signed Key is invalid"*
SSE 02	*"Format of Self Signed Key file is incorrect"*
SSE 03	*"SA Signed Data Server Certificate is invalid"*
SSE 04	*"Format of SA Signed DS Certificate is incorrect"*
SSE 05	*"SA Digital Certificate (X509) file is not available. A valid certificate can be obtained from the IHO website or your data supplier"*
SSE 06	*"The SA Certificate/Public Key is invalid. The SA may have issued a new public key or the ENC may originate from another service. A new SA public key can be obtained from the IHO website or from your data supplier"*
SSE 07	*"SA signed DS Certificate file is not available. A valid certificate can be obtained from the IHO website or your data supplier"*
SSE 08	*SA Digital Certificate (X509) file incorrect format. A valid certificate can be obtained from the IHO website or your data supplier*
SSE 09	*ENC Signature is invalid*
SSE 10	*Permits not available for this Data Server. Contact your data supplier to obtain the correct permits.*
SSE 11	*Cell Permit file not found. Load the permit file provided by the data supplier.*
SSE 12	*Cell Permit format is incorrect. Contact your data supplier and obtain a new permit file.*
SSE 13	*Cell Permit is invalid (checksum is incorrect). Contact your data supplier and obtain a new permit file.*
SSE 14	*Incorrect system date, check that the computer clock (if accessible) is set correctly or contact your system supplier.*
SSE 15	*Subscription service has expired. Please contact your data supplier to renew the subscription licence*
SSE 16	*ENC CRC value is incorrect. Contact your data supplier as ENC(s) may be corrupted or missing data.*
SSE 17	*Userpermit is invalid (checksum is incorrect). Check that the correct hardware device (dongle) is connected or contact your system supplier to obtain a valid userpermit.*
SSE 18	*HW_ID is incorrect format*
SSE 19	*Permits are not valid for this system. Contact your data supplier to obtain the correct permits*
SSE 20	*Subscription service will expire in less than 30 days. Please contact your data supplier to renew the subscription licence*
SSE 21	*Decryption failed no valid cell permit found. Permits may be for another system or new permits may be required, please contact your supplier to obtain a new licence*
SSE 22	*SA Digital Certificate (X509) has expired. A new SA public key can be obtained from the IHO website or from your data supplier.*
SSE 23	*Non sequential update, previous update(s) missing try reloading from the base media. If the problem persists contact your data supplier.*
SSE 24	*ENC Signature format incorrect, contact your data supplier*
SSE 25	*Viewer - The ENC Permit for this cell has expired. This cell may be out of date and MUST NOT be used for NAVIGATION*
SSE 26	*Viewer – Updates for this cell are missing and therefore out of date and MUST NOT be used for NAVIGATION*
SSE 27	*"This ENC is not authenticated by the IHO acting as the Scheme Administrator"*

SSE 01 must be returned when a self signed key (SSK) cannot be validated against the public stored as part of the SSK. The data server must check that its own SSK is valid before sending it to the SA. The SA will confirm that the date server SSK before returning the SA signed data server certificate.

SSE 02 must be returned if the SSK is wrongly formatted. That is elements of the SSK or characters are missing. The SA and data servers must complete this check.

SSE 03 must be returned if the SA signed data server certificate does not validate against the SA public key. This must be carried out by the SA before supplying it to the data server. The data server on receipt from the SA and the data client when authenticating the certificate in the ENC signature file prior to decryption.

SSE 04 must be returned if the SA signed data server certificate is wrongly formatted. This must be carried out by the data server on receipt from the SA.

SSE 05 must be returned if there is no certificate installed on the data client or the path to it cannot be found.

SSE 06 must be returned if the SA digital certificate (public key) does not validate against the following:

SA digital certificate will not validate against the SA public key.
The SA public key contained in the digital certificate will not authenticate against the signature contained in the ENC signature file. This could be a case of the certificate being invalid or an invalid or badly formatted signature.

SSE 07 must be returned if the SA signed data server certificate is not available to the data server for checking or is not present in the ENC signature file when the data client attempts to authenticate it.

SSE 08 must be returned if the SA public key held in the SA digital certificate is wrongly formatted or the certificate file is unreadable.

SSE 09 must be returned if the ENC signature element in the ENC signature file does not authenticate against the data server public key contained in the certificate element of the ENC signature file.

SSE 10 must be returned if there are no cell permits available for a particular data server corresponding to the exchange set being loaded.

SSE 11 must be returned if there are no permits installed on the system.

SSE 12 must be returned if the cell permits are formatted incorrectly.

SSE 13 must be returned if the calculated CRC of the cell permit does not validate against the CRC held in that cell permit. [Data Clients]

SSE 14 must be returned if the system date does not agree with the date obtained from any alternative, reliable date source, e.g. GPS. [Data Clients]

SSE 15 must be returned if the expiry date of the cell permit has an earlier date than that obtained from the validated system date. [Data Clients]

SSE 16 must be returned if the calculated CRC value of the ENC (after decryption and uncompressing) does not validate against the corresponding CRC value in the CATALOG.031 file. This also applies to the unencrypted signature, text and picture files. [Data Clients]

SSE 17 must be returned if the CRC contained in the userpermit does not validate against the calculated CRC of the extracted HW_ID. [Data Servers]

SSE 18 must be returned if the if the decrypted HW_ID extracted from the userpermit is incorrectly formatted. [Data servers]

SSE 19 must be returned if the HW_ID stored within the hardware/software security device cannot decrypt the cell permits being loaded or already installed on the system.

SSE 20 must be returned if the subscription licence is due to expire within 30 days or less.

SSE 21 must be returned if a valid cell key (decryption key) cannot be obtained from the relevant cell permit to enable the system to decrypt the corresponding ENC cell.

SSE 22 must be returned if the SA Digital Certificate (X509) has expired. That is if the ***"Valid to"*** date in the certificate is older than the validated system date.

SSE 23 must be returned

THIS SECTION TO BE COMPLETED AND VERIFIED

Page intentionally left blank

S-63 Annex A

Data Server Certificate Request Procedure

Page intentionally left blank

1 Purpose

The purpose of this procedure is to define the process for a Data Server to obtain a SA signed Data Server Certificate from the SA as defined by the IHO S-63 Data Protection Scheme Standard.

2 Responsibility

2.1 Need for Data Server Certificate

An organisation that encrypts and digitally sign ENC data as part of the IHO S-63 Data Protection Scheme will require a Data Server Certificate signed by the Scheme Administrator.

Users of encrypted and digitally signed ENC data (i.e. ECDIS systems to authenticate a signature and decrypt ENC information) do not need a Data Server Certificate. Agents or Distributors who will only provide ENC services supplied by a Data Server will not require a Data Server Certificate.

2.2 Hydrographic Offices and RENC Organisations

All Hydrographic Offices and RENC (Regional ENC Coordinating Centre) organisations only have to complete Part I of the attached form and include the required information to apply for a Data Server Certificate. A Data Server can only obtain one Data Server Certificate.

2.3 Non-Hydrographic Offices and Non-RENC Organisations

Other commercial organisations who wish to operate as Data Servers and encrypt and digitally sign ENC information compliant with the protection scheme can apply for a Data Server Certificate. Such organisations must get a Data Server, already a member of the protection scheme, to endorse the request and complete part II of the form. It is assumed that the Data Server providing the ENC data to the commercial organisation will endorse the request.

2.4 *International Hydrographic Bureau*

The IHB as Scheme Administrator has the sole responsibility to generate the Data Server Certificates compliant with internal procedures.

3 Definitions

Data Server: This is the term used to identify an organisation producing encrypted ENC data that is digitally signed and issuing Cell Permits to Data Clients (end-users).

Certificate: Certificates are digital documents attesting to the binding of a public key to an individual or organisation. They verify that a specific public key belongs to a specific organisation, in this case the IHO.

3.1 *References*

[1] IHO S-63 Data Protection Scheme, International Hydrographic Organisation
[2] IHO S-57 Transfer Standard for Digital Hydrographic Data, International Hydrographic Organisation

4 Procedure

This chapter defines the flow of information, responsibilities and detailed work instructions.

4.1 Completion of Forms and Attachments

A Data Server, who is already either a Hydrographic Office or a recognised RENC, who wishes to become a participant in the IHO S-63 Data Protection Scheme, is responsible for providing the following information to the IHB:

- A signed IHO Data Server Agreement
- A signed Certificate Request Form with Part I filled out
- The Data Server's Public Key
- The Data Server's Self Signed Certificate (SSK)

4.2 Need for Endorsement

All non-hydrographic offices and non-RENC organisations wishing to become a Data Server must have the Certificate Request Form endorsed by an existing Data Server who is already a member of the scheme.

4.3 Endorsing Organisation

The endorsing Data Server must complete Part II of the Certificate Request Form and return it to the non-hydrographic offices or non-RENC organisation.

4.4 Submission of Request to IHB

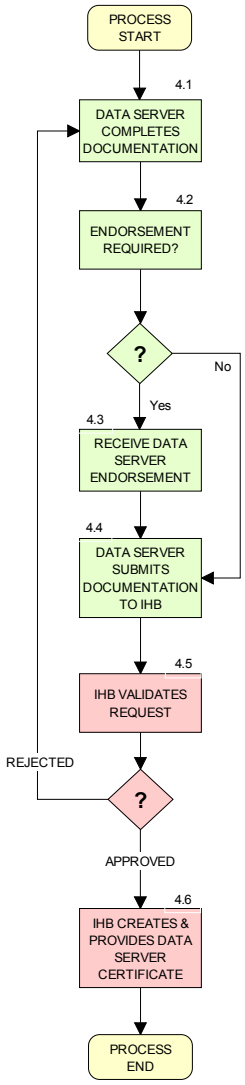
The Data Server is responsible for submitting the completed Request Form together with all other information listed in section 4.1 above to the IHB.

4.5 Validation of Certificate Request

IHB will validate the origins of the Certificate Request and authenticate the public key by contacting the Data Server. It will also ensure the need for a Data Server Certificate is applicable by contacting the endorsing Data Server. IHB will report discrepancies back to the originator.

4.6 Creation of Data Server Certificate

IHB is responsible for the authentication of the SSK created by the Data Server. If authentic the IHB then signs the Data Server public key to create the Data Server's Certificate, this is then supplied to the Data Server.



5 Quality Metrics

The IHB will archive the Data Server Request information and attachments compliant with internal procedures.

IHO S-63 Data Protection Scheme
Data Server Certificate Request Form

Ed.1-2003

Form to be returned to:
International Hydrographic Bureau
4, Quai Antoine 1er, B.P 445 - MC 98011 MONACO Cedex
Principality of Monaco
Tel: +(377) 93 10 81 00, Telefax: +(377) 93 10 81 40

Part I: To be completed by Data Server organisation

Organisation: .....
Address: .....
Address: .....
Address: .....
Postal number/place: ..... **Country**: .....
Tel: ..... **Fax**: ..... **Web**: .....

Administrative point of contact:	***Technical point of contact:***
Name:	**Name**: .....
Tel:	**Tel**: .....
E-mail:.....	**E-mail**: .....

Please verify the following information is included:
☐ All fields in Part 1 & 2 of this form are completed
☐ Data Server Public Key
☐ Data Server Self Signed Key (SSK)
☐ Signed IHO S-63 Data Server Agreement, or ☐ already available with IHB

Signed date: ..... **Name**: .....

Part II: To be completed by endorsing HO or RENC organisation

Organisation: .....
Contact name: .....
Tel: ..... **Fax**: ..... **E-mail**: .....

Part III: To be completed by IHB

☐ Form and attachments validated
☐ Signed Data Server Agreement, ref. …….
☐ Certificate created date: ..... Fileref: .....
☐ Certificate returned to Data Server

Signed date: ..... **Name**: .....

Page intentionally left blank

S-63 Annex B

Manufacturer Information Request Procedure

Page intentionally left blank

1 Purpose

The purpose of this procedure is to define the processes that an OEM has to undertake to become a participant in the IHO S-63 Data Protection Scheme. To participate, OEMs will require their own unique M_ID and M_KEY values. These are supplied by the SA as defined by the IHO S-63 Data Protection Scheme so that OEMs can decrypt S-63 encrypted ENCs.

2 Responsibility

2.1 OEMs

Only OEMs that develop Data Client applications need a unique M_ID and M_KEY value. IHB as the Scheme Administrator will share this information with all the Data Servers participating in the scheme. An OEM will only be issued with one M_ID and M_KEY pair.

The M_ID and M_KEY values will be returned to the Scheme Administrator if the organisation ceases trading or no longer supports an application that accesses and displays S-63 encrypted ENCs. Data Servers will be informed of such instances so that no new licences are issued for that particular manufacturers system.

There is no need for Data Client to have access to the M_KEY value because it is securely built into the end-user application (e.g. dongle) and supplied to Data Clients in an encrypted form known as a userpermit.

2.2 International Hydrographic Bureau

The IHB as SA has the sole responsibility for generating the M_ID and M_KEY values and supplying them to OEMs and distributing among Data Servers.

3 Definitions

M_ID:	Manufacturer Identification
M_KEY:	Manufacturer Key
OEM:	Original Equipment Manufacturer
Userpermit:	A 28 character alphanumeric string containing the Data Client's HW_ID encrypted with the manufacturer's M_KEY and containing the M_ID.
Dongle:	A hard lock device that contains the HW_ID of the Data Clients system.

3.1 References

[1] IHO S-63 Data Protection Scheme, International Hydrographic Organisation
[2] IHO S-57 Transfer Standard for Digital Hydro-graphic Data, International Hydrographic Organisation

4 Procedure

This chapter defines the flow of information, responsibilities and detailed work instructions.

4.1 *Completion of Request Form*

The OEM is responsible for completing all information in Part I of the attached M_ID and M_KEY request form. The IHO may wish to require further documentation such as Confidentiality Agreements – these are not detailed here.

Note that an OEM can:

- Only be assigned one M_ID and M_KEY pair
- Must return the information to the Scheme Administrator (SA) if it stops trading or does not deliver products authenticating signatures or no longer has a need to decrypt ENC information.

4.2 *Verification of Request Form*

The SA verifies that all information in Part I of the form is completed, or provide information to OEM about missing information.

4.3 *Verification of Signed Confidentiality Agreement*

The SA verifies that a signed Confidentiality Agreement is included with the request or already available in the IHB archive. If an Agreement is not available, inform the OEM about the mandatory need for a signed Agreement.

4.4 *Confirm Successful Testing with S-63 Test data*

Verify that the OEM has completed successful testing of application with the available IHO S-63 test data. If not, request the OEM to complete the defined test procedure before M_ID and M_KEY are provided.

4.5 *Check OEM has no Current M_ID and M_KEY*

Verify the OEM has no previously assigned M_ID and M_KEY. If not, inform the OEM about the problem.

4.6 *Creation of M_ID and M_KEY*

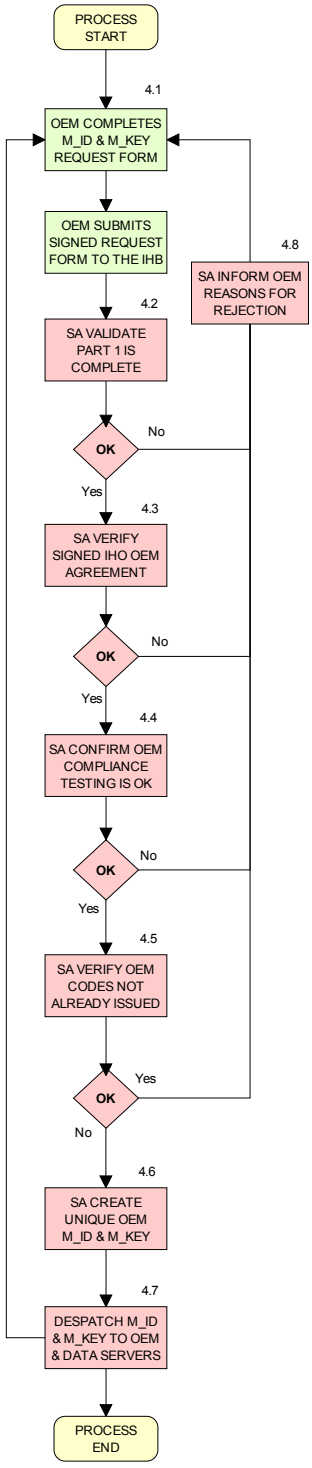
The SA assigns the OEM an available and unique M_ID and M_KEY combination.

4.7 *Inform About New M_ID and M_KEY*

The SA informs the OEM about its M_ID and M_KEY. The SA informs all registered Data Servers about the new M_ID and M_KEY information.

4.8 *Inform OEM about Problem with Request*

The SA informs the OEM about specific problems with the request and requests updated information to be provided before a M_ID and M_KEY can be assigned. Further processing of the request is terminated.



5 Quality Metrics

The IHB will archive the Request form and all relevant information compliant with internal procedures.

IHO S-63 Data Protection Scheme
M_ID and M_KEY Request Form

Ed.1-2003

Form to be returned to:
International Hydrographic Bureau
4, Quai Antoine 1er, B.P 445 - MC 98011 MONACO Cedex
Principality of Monaco
Tel: +(377) 93 10 81 00, Telefax: +(377) 93 10 81 40

Part I: To be completed by OEM organisation

Organisation: .....
Address: .....
Address: .....
Address: .....
Postal number/place: ..... **Country**: .....
Tel: ..... **Fax**: ..... **Web**: .....

Administrative point of contact:	***Technical point of contact:***
Name:	**Name**: .....
Tel:	**Tel**: .....
E-mail:.....	**E-mail**: .....

Please verify the following information is included:

☐ All fields in Part 1 of this form are completed
☐ Signed IHO S-63 OEM Agreement, or ☐ already available with IHB
☐ Completed successful testing of application with the M_ID and M_KEY provided with the S-63 test dataset

Signed date: ..... **Name**: .....

Part II: To be completed by IHB

☐ Verify Part 1 is completed
☐ Signed OEM Agreement available, ref. .....
☐ Verify OEM does not have a previously issued M_ID and M_KEY
☐ Assigned M_ID: ..... M_KEY: .....
☐ M_ID and M_KEY returned to OEM and all registered Data Servers

Signed date: ..... **Name**: .....

Page intentionally left blank

S-63 Appendix 1

Data Protection Scheme Test Data

Page intentionally left blank

Important Notice: S-63 Appendix 1 includes test data which are provided separately as compressed (ZIP) files (see S-63 page on the IHO website – www.iho.shom.fr/ECDIS/). Embedded in the ZIP file is a document "Test Data Implementation Guide" providing instructions on how to use the test data. The text below provides a brief presentation of Appendix 1.

1 Introduction

The S-63 Appendix 1 defines a recommended set of test definitions and test data which can be used by developers of Data Server and Data Client applications to understand the security constructs defined in S-63 and test if their application is compliant with the standard. It includes a "Test Data Implementation Guide" which is provided along with the test data.

The S-63 Appendix 1 will be maintained by the IHO DPSWG. More test data can be included in the future based on user feedback to provide a complete test platform to verify correctness and compliance with the standard, or for end-user applications to identify erroneous situations. The current version of the document provides a complete test sample for compliance testing.

The associated "Test Data Implementation Guide" will be maintained independent of the IHO S-63 main document and new versions will be published on the IHO website.

Questions related to the use of the test data can be posted at the *Open Ecdis Forum* (www.openecdis.org).

2 Organisation of the Test Definitions and Test data

2.1 Test Definitions

The test definitions offers high level functional tests which are recommended to test for compliance with all security constructs defined in S-63. It does not replace unit testing in software development, but offer structured input to functional software testing.

The test definitions are organised in functional categories and defined in chapter 3 of the "Test Data Implementation Guide". Test definitions for the Scheme Administrator functionality has not been included in the document since only the IHB will require these test scenarios.

Each test definition indicates if the test is applicable for Data Server or Data Client applications. Note that a test is relevant for all applications if the type of application is omitted.

There are test definitions for both good and erroneous test conditions to ensure a robust application and reflect operational conditions.

Note that the IEC will be responsible for defining applicable ECDIS type approval tests which will complement this document.

2.2 Test data

A range of test data has been developed to support the test definitions. The features of each test data set is defined in chapter 4 of the "Test Data Implementation Guide".

All the test data is organised in a ZIP file and will extract into a directory structure where each test data will be located in a separate directory. Note that some of the test data sets are used in multiple test definitions.

Note that the test data can also be used by developers for unit testing or other test situations for their application.

2.3 Conditions of Use for the Test data

The ENC information (the material) included in the test data has been made available to the recipient solely for the purpose of testing their application and verifying compliance with the S-63 standard. The material is supplied under the conditions shown below. If the recipient does not agree to bound by these conditions then the material should not be used and it should be destroyed.

2.3.1 Conditions of Release

The material supplied is protected by the copyright of the national Hydrographic Office. No part of the supplied material may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying or otherwise except as required to fulfil the purpose described above.

The material is NOT to be used for navigation.

When the material is no longer required to fulfil the purpose, it and any working copies, are to be destroyed.

2.3.2 Disclaimer

Whilst the IHB and the Hydrographic Offices have used its best endeavours to ensure that the material is suitable to fulfil the purposes, they offer no warranty guarantee or other assurances to the effect that it will meet the requirements. The IHB and the Hydrographic Offices will accept no liability for any damage or loss of any nature rising form its use. The material supplied is used entirely at the recipient's own risk.

S-63 Appendix 2

Large Media Support

Edition 1.1

Page intentionally left blank

1 Introduction

Until recently the majority of ECDIS/ECS only had the capability to load ENC Exchange Sets (ExSets) from CD-ROMs. However, it is becoming increasingly common for new OEM hardware to be delivered with DVD drives or other Large Media Support[14] (LMS). The inclusion of this media support now offers data servers the potential to include more ENC data on a single media.

Several issues have emerged during the operation of the existing S-63 encrypted ENC services using edition 1.0 of the standard. Not least of these is the fact that providing large exchange sets has resulted in unacceptable load times to the ECDIS/ECS. This is one of the principle reasons why data servers did not provide services that included single exchange sets spanning multiple CD-ROMs.

To store a single ENC exchange set on a mass storage device such as a DVD or USB that was similar in size to that stored on CD-ROM, would be an inefficient use of the media and the available memory. This being the case it would make sense to store multiple exchange sets on the same media, each about the same size as those currently stored on CD-ROM. Since this method of storage is not defined in the IHO S-57 Product Specifications or Edition 1.0 of S-63 a new configuration will have to be specified.

When designing the media structure the following considerations have been taken into account:

- ENC Services may be provided across multiple media sets
- ENC Services may contain data from more than one Data Server
- Suitable files must be provided so that manufacturer systems can manage and import S-63 encrypted ENCs in an efficient and expeditious manner and create intuitive systems which manage multiple pieces of media easily.

2 Media Overview

The following section gives a high level view of how the data will be structured on the media. It also outlines how the S-63 exchange set structure has been modified for large media support. A high level diagram of this is provided in Annex A of this appendix. Detailed information relating to the content and format of these folders and files are provided later in this appendix.

2.1 Media Types

There will be two types of media, ***"BASE"***, containing one or more Base Exchange sets and, ***"UPDATE"***, containing the weekly ENC updates, these may be contained in one or more exchanges sets on the update media.

It was considered that because of the increased capacity which these types of media offer it could be possible to re-issue base ExSets on the update media and conversely weekly updates on the base media.

2.2 Media Folder and File Structure

All exchange sets are located in the root directory of each media each within their own specific sub-directory. The configuration of all exchange sets are the same as outlined in 7.5.1 of the main document with the one notable exception. The ***"INFO"*** folder that includes the ***"PRODUCTS.TXT"*** file will no longer be stored in the root directory of the exchange set(s) but in the root directory of the media.

The ***"INFO"*** will continue to be used by data servers to include additional files unique and specific to their S-63 encrypted ENC services. Note that any data server specific files stored in this folder must be named in such a way that they **DO NOT** conflict with the S-63 file naming convention.

[14] Large Media Support can also be referred to as Mass Storage Devices.

2.2.1 Additional Media File

An additional file named ***"MEDIA.TXT"*** will be included on each piece of media to assist data clients in managing multiple exchange sets on the same media and across multiple media sets. It will also enable data clients systems to prompt users to insert the relevant media by including a machine readable string in each record referring to each piece of media. A more detailed explanation of the format of the ***MEDIA.TXT*** file is provided later in section 3.

2.3 Media Identification

There must be a method for differentiating between services provided on CD and those supplied using large media support. The first differentiator is the volume ID of the media (see section 32.32.3.1). This will identify the use of large media format and notify the data client of the expected folder and file structure.

A further indication is the presence of the new MEDIA.TXT file is in the root directory of the media is a further indication that an ENC service is being provided using large media support.

2.3.1 *Media Labelling*

For large media support the media labelling convention will be similar to that used in the IHO S-57 product specification. Instead of "V01X01", where "V" stands for "Volume", the letter "M" for "Media" will be substituted.

The volume label for large media support will also indicate how many media sets there are in a service. Therefore if there are three media sets they will be labelled as follows:

M01X03 [Media set 1 of 3]
M02X03 [Media set 2 of 3]
M03X03 [Media set 3 of 3]

NOTE: This only identifies the number of media sets in an ENC service and does not imply that this is a single exchange set covering multiple media sets. This purpose of this naming convention is to assist data clients to identify the media where licenced cells are located.

3 Media File Formats

3.1 Product Listing (PRODUCTS.TXT)

For ***"Base Media"*** the ***"PRODUCTS.TXT"*** file will contain records for all cells held on that particular media. The header as defined in section 6.2.2 of the main document will be labelled as ***"FULL"*** if there is only one media in a particular service. However if there is more than one media this will be labelled ***"PARTIAL"***. A ***"FULL"*** products listing will always be provided on the ***"Update Media"*** with records of all cells in a data server's service.

It is important that ECDIS/ECS manufacturers manage these records carefully; ***"PARTIAL"*** product listings must be merged with the ***"FULL"*** listing stored within the system. It must be noted that the system may contain product information from more than one data server. Therefore it is important not to overwrite ***"FULL"*** listings unless they are stored independently according to data server.

3.2 Media Listing (MEDIA.TXT)

This is a new file designed to manage services supplied using large media support. It is located in the root directory of the base and update media and contains information relating to all media in a data server's service and the exchange sets contained on the media. The main purpose of this file is as follows:

- To provide data clients with a means to manage the import of a data server's service that supports large media.

- To provide information to allow data clients to manage multiple media sets.
- To provide machine readable information so that data clients can make the import process more intuitive for the end user.

NOTE: The latest update media will always contain the most up to date status of the base exchanges sets in a data server's service. This can be used to check the latest base exchange set has been installed. Further details on the structure and format of this file are outlined below.

5.1.1 Header Format

The purpose of the MEDIA.TXT header is similar to that of the SERIAL.ENC file stored with the exchange set. It is used to manage the media installation by identifying the following:

- The media service provider
- The media date and week of issue
- The media number and media type
- Machine readable media name to display to users

The format of the header is provided in the table below:

Field ID	Domain	Bytes	Range
Data Server ID	Character	2	Any pair of alphanumeric, e.g. **PR**
Week of Issue	Character	10	Any ASCII Characters, e.g. **WKNN_YY**
Date of Issue	Date	8	**YYYYMMDD**
Media Type	Character	10	**BASE** or **UPDATE**
Media Label ID	Character	10	**M[01-99]X[01-99]**
End of record delimiter	hexadecimal	2	**CR/LF [0x0D0A]**
Media ID	Character	3	For example, **M1**, **M2** or **M11.**
Machine Readable Media Name	"Character"	>100	Text string contained within quotation marks
Regional Information [Optional]	"Character"	>100	Text string contained within quotation marks
End of record delimiter	hexadecimal	2	**CR/LF [0x0D0A]**

Commentaire [jp3] : Do we have to use this!!! Not sure why it was put in in the first place but it seems silly to replicate it further… A simple CRLF would be fine and easier for OEMs

Commentaire [jp4] : Do we have to use this!!! Not sure why it was put in in the first place but it seems silly to replicate it further… A simple CRLF would be fine and easier for OEMs

Example:

```
GBWK27_07   20070621BASE    M01X02
M1,"UKHO BASE MEDIA 1","Europe, Africa, and Middle East"
```

3.2.2 Media Record Format

The ***"MEDIA.TXT"*** file also contains a list of records that identifies all exchange sets in a data server's service and the destination media where they can be located. Its purpose is to provide data clients with a means of managing the import of encrypted ENCs across multiple media sets and provide machine readable information so that the data client can prompt end users to load the appropriate media.

The ***"MEDIA.TXT"*** file stored on the **UPDATE** media will always contain a **FULL** list of media sets contained in a data server's service. It will also carry the date when the media was last issued, this way the ECDIS/ECS can always validate whether it holds the latest information.

The ***"MEDIA.TXT"*** file stored on the **BASE** media will contain a list of those exchange sets stored on the media. It will **NOT** contain information about the other volumes in the service.

Field ID	Domain	Bytes	Range	Notes (see below)
Media/Exchange Set Location	Character	3	**M1** to **M99;B1** to **B99** e.g. **M2;B7** [Media 2, Base ExS 7]	1

Exchange Set Issue Date	Date	8	**YYYYMMDD**, e.g. 20070621	2
Media Set Number [Long Name]	Character		"Any ASCII Characters"	3
Regional Information [Optional]	Character		"Any ASCII Characters"	4
Reserved Field	Character			5
Comments Field	Character			6

Example:

Notes:
1. This field identifies on what media the base or update exchange set is located.
2. The ExSet Issue Date. This is the date when an ExSet is issued or re-issued[15] on the base or update media. Although it may be more practical to re-issue all ExSets on a particular media simultaneously there may be occasions when the media is re-issued with just one ExSet re-issued. Data Clients may use this date to validate the status of the currently installed cells from the update media.
3. This is a machine readable text string that Data Clients can use to prompt end users to load the appropriate media.
4. This is an optional machine readable text string that can used by data clients to display additional information relating to the regions/producer nations on a particular media..
5. Future Use
6. Additional comment information

The update media ***"MEDIA.TXT"*** file will always contain the latest issue dates and information for all base media exchange sets in a media set. *Although provision has been made to have more than one update ExSet on the update media, it is not recommended for the reasons mentioned in section 4. However, if there are more than one then this can be managed by the entries in the PRODUCTS.TXT and MEDIA.TXT file on the update media.*

Example of a complete **MEDIA.TXT [UPDATE]**:

```
GBWK28_07   20070628UPDATE    M01X01
U1,"UKHO UPDATE MEDIA"
M1;B1,20070614,"Europe",,
M1;B1,20070614,"UKHO BASE MEDIA 1","Europe, Africa and Middle East",,
M1;B2,20070614,"UKHO BASE MEDIA 1","Europe, Africa and Middle East",,
M1;B3,20070621,"UKHO BASE MEDIA 1","Europe, Africa and Middle East",,
M2;B4,20070517,"UKHO BASE MEDIA 2","North and South America",,
M2;B5,20070517,"UKHO BASE MEDIA 2","Morth and South America",,
M3;B6,20070405,"UKHO BASE MEDIA 3","Far East and Australasia",,
M3;B7,20070405,"UKHO BASE MEDIA 3","Far East and Australasia",,
M1;U1,20070628,"UKHO UPDATE MEDIA 3","Europe",,
M1;U2,20070628,"UKHO UPDATE MEDIA 3","Rest of the World",,
U1,20070628,"Updates for Week MM/YY",,
```

Example of complete **MEDIA.TXT [BASE]** file on:

```
GBWK27_07   20070621BASE      M01X03
M1,"UKHO BASE MEDIA 1","Europe, Africa, and Middle East"
M1;B1,20070614,"Base Dataset 1","Europe",,
M1;B2,20070614,"Base Dataset 2","Africa",,
M1;B3,20070621,"Base Dataset 3","Middle East",,
```

4 Media Management (Data Servers)

The issue and re-issue of base media is very much at the discretion of the data server. However, to prevent the continual renewal of base media it is recommended that individual exchange sets are not issued independently of one another on the same media. However, there may be occasions when this

[15] A re-issued exchange set is one that contains the entire base ENCs (plus updates) assigned to it plus any new editions and updates issued since the previous re-issue.

is necessary, e.g. the introduction of ENCs from a new country or essential management of the update exchange set.

If a data server is operating a two tier service, e.g. they support both a CD-ROM and DVD services. The content of the Base and Update Exchange sets will be the same in both services. It may well be that data server's issue the Base Exchange sets on DVD and the weekly update on CD. This would keep the production costs to a minimum.

5 Media Management (Data Clients)

As the volume of ENCs continues to grow a more intelligent and "smarter" method of loading them into ECDIS/ECS is required. Since most customer only purchase a subset of all available ENCs it would seem prudent to base the import of S-63 encrypted ENCs directly against the customers permit holdings. The following numbered points are provided to illustrate the recommended steps for importing data.

1. Insert, read and validate the permit file
2. Insert the update media
3. Read the ***"FULL"*** products listing
4. Identify and flag which cells have valid permits
5. Indentify the base cell location of each cell with a valid permit
6. Data client prompts user to install the appropriate base media
7. When all licenced cell are loaded from the base media the user is prompted to insert the update CD bring the whole collection up to date and complete the ENC loading cycle.

5.1 Media Warnings

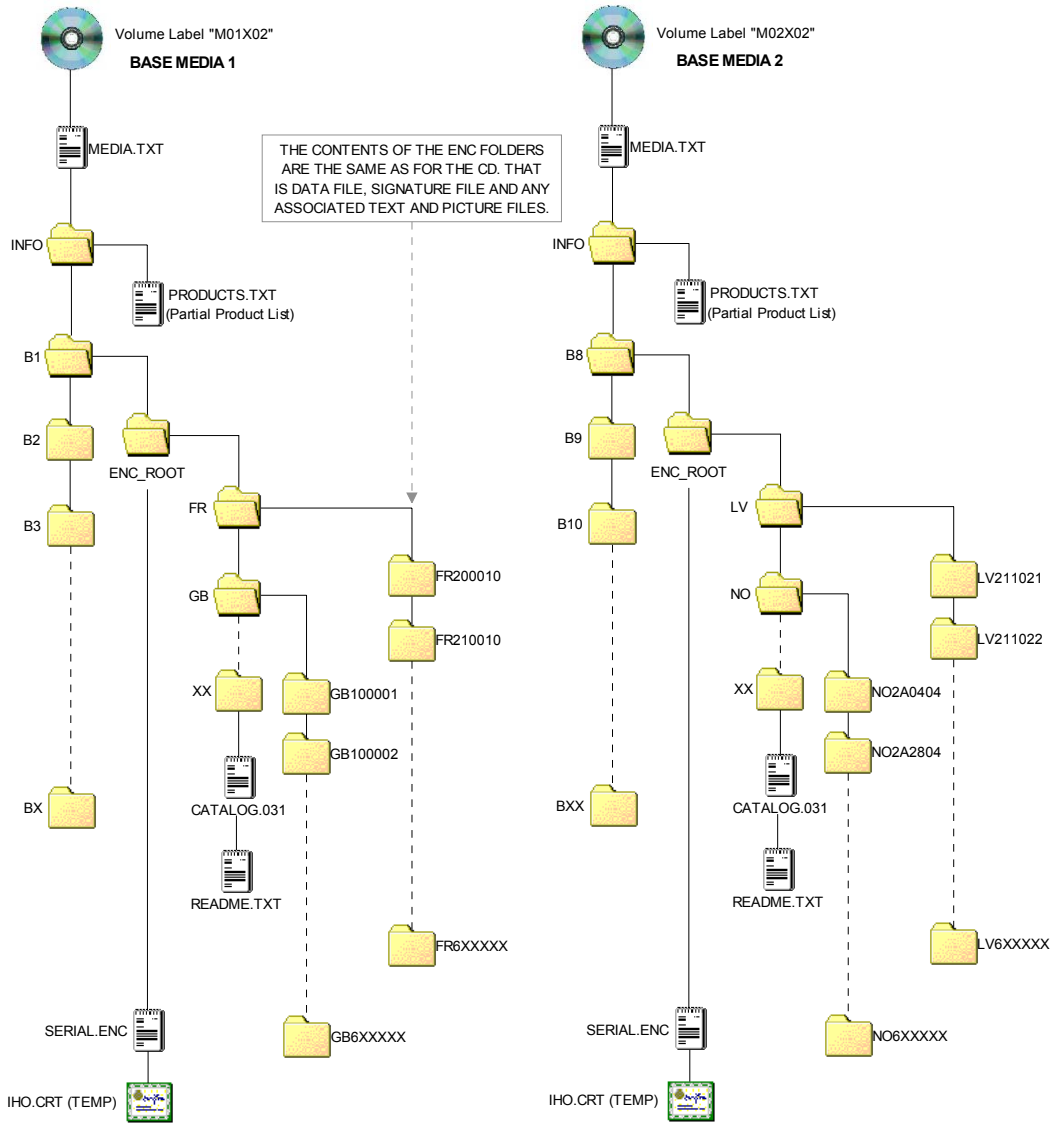
When the weekly update media is loaded data clients must check that the issue date of all installed exchanges sets are current and up to date. The latest update media will always contain the latest issue dates and week numbers of each exchange set in a service in the ***"MEDIA.TXT"*** file.

If the ECDIS/ECS does not have the latest base media loaded a warning must be given instructing the user as follows:

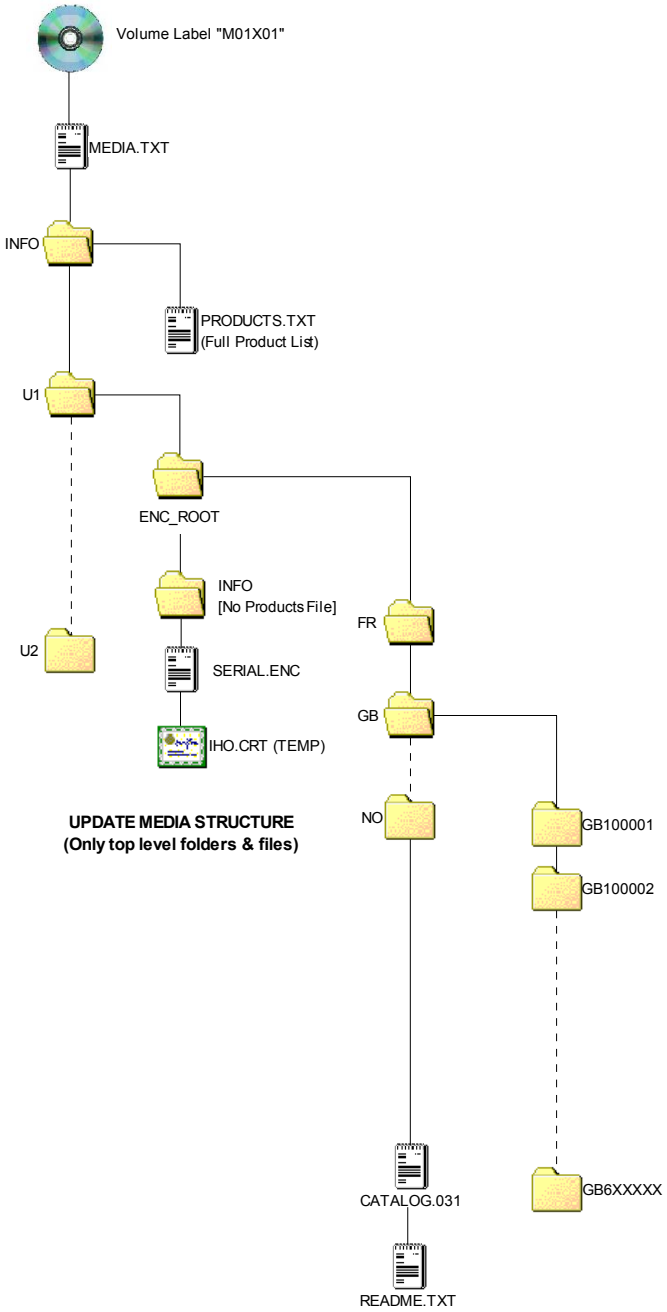
"UKHO Base Media x has been re-issued please load the latest Base Media x with an issue date of YYYYMMDD"

ANNEX A

The diagram below is for illustrative purposes only and outlines the top level folder and file structure that must be used by data servers when supplying S-63 encrypted ENC services utilising large media support. However, it is possible that the structure under each ENC_ROOT folder of each exchanges set may vary between data servers.



LARGE MEDIA SUPPORT FOR S-63 ENCRYPTED ENC SERVICES
BASE MEDIA FOLDER AND FILE STRUCTURE


Volume Label "M01X01"
MEDIA.TXT
INFO
PRODUCTS.TXT
(Full Product List)
U1
ENC_ROOT
INFO
[No Products File]
SERIAL.ENC
IHO.CRT (TEMP)
U2
FR
GB
NO
GB100001
GB100002
GB6XXXXX
CATALOG.031
README.TXT
UPDATE MEDIA STRUCTURE
(Only top level folders & files)